January 2021

Developing A Machine Learning Based Approach For Fractured Zone Detection By Using Petrophysical Logs

Haleh Azizi

Developing a Machine Learning Based Approach for Fractured Zone Detection by Using Petrophysical Logs

By

Haleh Azizi

Bachelor of Science, Mining Engineering, Shahrood University of Technology, 1999

Master of Science, Geophysics, Seismic, Science and Research Branch, Azad University, 2005

PhD, Geophysics, Seismic, Science and Research Branch, Azad University, 2010

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota

May

2021

ProQuest LLC.
789 East Eisenhower Parkway
P.O.Box 1346
Ann Arbor, MI 48106 - 1346

Name: Haleh Azizi

Degree: Master of Science

This document, submitted in partial fulfillment of the requirements for the degree from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

DocuSigned by:
Dr Hassan Reza
Dr. Hassan Reza

DocuSigned by:
Dr. Kouhyar Tavakolian
Dr. Kouhyar Tavakolian

DocuSigned by:
Wen-Chen Hu
Dr. Wen-Chen

Name of Member 4 - delete if not needed

Name of Member 5 - delete if not needed

Name of Member 6 - delete if not needed

This document is being submitted by the appointed advisory committee as having met all the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

DocuSigned by:
Chris Nelson
Chris Nelson
Dean of the School of Graduate Studies

2/10/2021
Date

PERMISSION

Title:	Developing a Machine Learning Based Approach for Fractured Zone Detection by Using Petrophysical Logs
School	Electrical Engineering and Computer Science
Degree	Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in his absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Haleh Azizi
January 25, 2021

Table of Content

Content **Page**

Content **Page**

Content **Page**

List of Figures

Title **Page**

List of Tables

Content **Page**

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Dr. Hassan Reza, Dr. Wen-Chen Hu and Dr. Kouhyar Tavakolian for being members of my committee. I am so thankful for their guidance and support during my time in the master's program at the University of North Dakota. I would also like to thank all the faculty and staff at the School of Electrical Engineering and Computer Science for the help they provided during my graduate career. Finally, I would like to thank my family for their love and support while on this academic journey.

To my husband and my two sons,

I dedicate this master's thesis to you. Your everlasting love and support continue to push me toward achieving my goals and untapped potential. Without you this would not have been possible.

I love you.

Abstract

Oil reservoirs are divided into three categories: carbonate (fractured), sandstone and unconventional reservoirs. Identification and modeling of fractures in fractured reservoirs are so important due to geomechanical issues, fluid flood simulation and enhanced oil recovery.

Image and petrophysical logs are individual tools, run inside oil wells, to achieve physical characteristics of reservoirs, e.g. geological rock types, porosity, and permeability. Fractures could be distinguished using image logs because of their higher resolution. Image logs are an expensive and newly developed tool, so they have run in limited wells, whereas petrophysical logs are usually run inside the wells. Lack of image logs makes huge difficulties in fracture detection, as well as fracture studies. In the last decade, a few studies were done to distinguish fractured zones in oil wells, by applying data mining methods over petrophysical logs.

The goal of this study was also discrimination of fractured/non-fractured zones by using machine learning techniques and petrophysical logs. To do that, interpretation of image logs was utilized to label reservoir depth of studied wells as 0 (non-fractured zone) and 1 (fractured zone). We developed four classifiers (Deep Learning, Support Vector Machine, Decision Tree, and Random Forest) and applied them to petrophysics logs to discriminate fractured/non-fractured zones. Ordered Weighted Averaging was the data fusion method that we utilized to integrate outputs of classifiers in order to achieve unique and more reliable results. Overall, the frequency of non-fractured zones is about two times of fractured zones. This leads to an imbalanced condition between two classes. Therefore, the aforementioned procedure relied on the balance/imbalance data to investigate the influence of creating a balanced situation between classes.

Results showed that Random Forest and Support Vector Machines are better classifiers with above 95 percent accuracy in discrimination of fractured/non-fractured zones. Meanwhile, making a balanced situation in the wells by a higher imbalance index helps to distinguish either non-fractured or fractured zones. Through imbalance data, non-fractured zones (dominant class) could be perfectly distinguished, while a significant percentage of fractured zones were also labeled as non-fractured ones.

Keywords: Decision Tree, Deep Learning, Oil Reservoir, Ordered Weighted Averaging, Random Forest, Support Vector Machine

1

Introduction

1.1 Background and Motivation

Fractures have a significant effect on oil/gas production in fractured reservoirs. More than 60 (oil) and 40 (gas) percent of the world's reserves are held in carbonate rocks [1]. Oil/gas extraction from carbonate reservoirs is challenging and significantly depends on the fractures. Hence, developing a reliable fracture model in fractured reservoirs is vital and their recovery factor is low [2]. Actually, a reliable fracture model could be useful to define a more effective oil production strategy, to increase the recovery factor. Economically, a one percent increase in recovery factor in giant reservoirs, means millions of barrels more oil production, consequently, trillions of dollars more benefits.

To study the rock fractures, which have a long genesis history, data must be gathered. Fracture data could directly or indirectly be gathered. Direct fracture data gathering in exposed rocks in tunnels, galleries, and cores is possible. In direct-based, different fracture properties could be gathered, e.g. dip, dip direction, continuity, spacing, roughness, filling, filler, and aperture.

Seismic sections, petrophysical logs (PLs), well tests, mud lost and image logs are well-known sources for indirect fracture data gathering

. Naturally, fewer fracture properties in indirect-based methods could be gathered.

Image logs are solely tools for fracture detection in oil wells. Various valuable research studies were done for fracture detection using other tools e.g. seismic data, tectonically simulations, well tests, and PLs. PLs contain the highest resolution among the aforementioned list and they might be the best dataset for fracture detection. In the current study, an algorithm would be presented to identify fractured zones from PLs.

1.2 Problem Statement

Although several approaches were presented for fractured zone detection, not one is reliable except image logs.

Seismic data, well tests, mud lust data, petrophysical well logs and core description were direct and indirect procedures that were used for fracture detection [3, 4, 5, 6, 7], each associated with serious shortcomings. Low resolution of seismic data, well tests and mud lusts, as well as low core recovery, especially in fractured zones and lack of orientation of cores are just examples of shortcomings. The technology of image logs has been the ultimate solution [8]. Resolution of image logs are in the range of 1 mm, absolutely suitable for fracture detection. The problem is that the image log is a technology that was developed during the last four decades; therefore, in all wells that were previously drilled, image logs are not available. Meanwhile, image logs are expensive, and were not run in the majority of wells. For instance, in 8 wells of studied giant oil reservoirs, with the size about 38 mi by 7 mi (about 13 times the size of Grand Forks county), image logs were run; while, in more than 450 wells full set petrophysical logs (PLs) were run.

It should be considered that fracture simulation is an object based modeling procedure and it needs more data to achieve a reliable model. Also, fractures are channels for fluid flow, so their models

were usually used for fluid flow simulation, which is a dynamic modeling, again associated with high errors. Therefore, fracture simulation by using image log data is not recommended.

In last 15 years, researchers have captured machine learning/data mining techniques to detect fractured zones from PLs. Those attempts were rewarding. Fourier and wavelet transform, classification and data fusion techniques were mostly used to do that. The approach of the current thesis is similar. Two novelties were considered for this research: 1. more powerful classifiers will be utilized, and 2. the effect of making a balance between fractured zones and non-fractured ones in discrimination between those zones will be investigated.

1.3 Purpose of the Thesis

Applying different signal processing as well as data mining techniques to discriminate fractured/non-fractured zones has been an approach that was widely used during the last decade. Overall, the results have been acceptable. Integration of signal processing, classification, and data fusion with generalized accuracy of about 72 percent [9, 10] have been the best results that were reported. The impact of signal processing over discrimination of fractured/non-fractured zones were insignificant [9, 10]. Therefore, it is ignored, and classification/data fusion were data mining techniques utilized in current research. Decision tree, random forest, support vector machine, and deep learning are the classifiers that were used for the first time for this purpose. Ordered weighted averaging, which has been previously utilized, is the selected data fusion method.

Usually, the frequency of non-fractured zones is more than fractured ones [3]; therefore, the database for the two classes are imbalanced. In the current study, the impact of unbalancing, which was ignored in previous studies, is investigated. To do that, the whole of the procedure was

repeated over balanced data, and a comparison between results achieved from balance/imbalance data are reported.

1.4 Organization of the Thesis

In chapter 1 a brief introduction and general overview about the topic of the thesis are presented. Chapter 2 contains a brief introduction to fractures as well as Asmari formation and its importance as the main reservoir of oil production. Host rock of building dams in Iran is also discussed in chapter 2. Asmari is a carbonate-fractured rock; hence, the factors control the fracture's genesis and their behavior is described too. Various research studies were conducted concerning fractures in Asmari, some of which are briefly reviewed. A review over fractured reservoirs is the title of the main section of chapter 2: a literature review. This chapter helped to know the history as well as state of the art research studies that were conducted for fractured zone detection. The approach and novelties of the thesis are backed up by the literature review, and in fact are a modification over the most successful procedures. Materials and methodology are reported in chapter 3. A description of image logs and then a brief introduction about the studied reservoir, location of the wells over UGC map, and a list of available PLs, are initially presented in chapter 3. Five preprocesses have been done over PLs, which are described. Interpretations of image logs were used to develop a log, called a fracture log. The procedure and a sample of achieved log are presented in chapter 3. Knowledge about the effect of fractured zones over PLs helps to log selection for classification procedures, which is the title of the next section in chapter 3. Following the proposed procedure for fractured zone detection, four classifiers and a data fusion method are used. Utilized classifiers are: decision tree, random forest, support vector machine, and deep learning, which are described in chapter 3. Ordered weighted averaging, the utilized data fusion

method, is the last method, introduced in chapter 3. Results and discussions are presented in chapter 4. Statistical studies supported by 1D, 2D, and 3D studies are done to show the importance of using more logs for fracture detection, as well as PLs selection. An imbalance index and then a selection procedure is defined, to make a database, its fractured/nonfractured zones are balanced. All fractured zone detection procedures are done over balanced and imbalanced datasets. It helped to investigate the effect of balancing on discrimination between fractured/non-fractured zones. Classification of individual wells by using decision tree, random forest, support vector machine, and deep learning helped to figure out the accuracy and precision of fractured/non-fractured zone discrimination as well as classifier selection for the generalization step. In the generalization step, other wells databases are utilized for fractured zone detection in each well. Random forest and support vector machine are the selected classifiers used in the second step. Results of classifications are integrated by ordered weighted averaging data fusion method, in order to achieve a robust, reliable approach for fractured zone detection. Comparisons between results over balanced and imbalanced data are presented in each step in chapter 4. A summary of contributions, the limitations of current research and the possible future directions and extensions are presented in chapter 5.

2

A Review of Fractured Reservoirs

2.1 Introduction

About half of world's reservoirs are carbonate, consequently, they are fractured [11]. Fractures have a significant effect over oil well stability and fluid flow in fractured reservoirs, which shows the importance of fracture detection and characteristics of its properties.

Rocks, unlike other materials, are not homogeneous and isotropic; on the contrary, a majority of them contain discontinuity. Therefore, rock bulks are in fact a series of stones discriminated by discontinuities.

Mechanically, each weak surface in the rock with low shear strength is called a discontinuity. Fractures are a group of discontinuities concluded of fragile failure. Joints and Faults are samples of fractures.

Fractures are planar structures sometimes along with shear displacement and sometimes just rock blocks get away from each other. Different factors affect fracturing, consequently, various fracture types could be achieved e.g. inductive, shrinkage, regional, or tectonical fractures.

All kinds of natural rock fractures have a significant effect on the stability of engineering structures, water reservoirs, and mineral deposits. They also have a significant effect on fluid flow as well as the geomechanical behavior of oil and gas reservoirs.

In an overall view, oil and gas reservoirs could be categorized in conventional and unconventional reservoirs. Conventional reservoirs could also be classified into sandstone and carbonate reservoirs. In Figure 2.1 the locations of mentioned reservoirs in the world are displayed. Also, in Figure 2.2 different types of carbonate reservoirs are displayed.

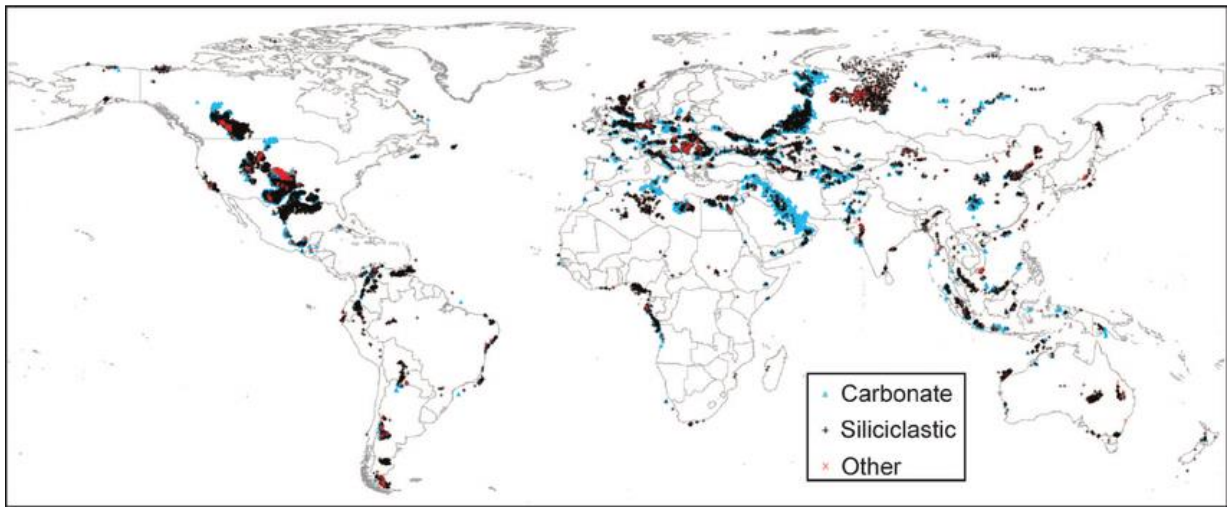


Figure 2.1 Location of carbonate, siliciclastic and unconventional reservoirs in the world [1].

2.2 Asmari Formation

The Oligocene–Miocene Asmari Formation is a thick sequence of shallow-water carbonates of the Zagros Basin. Asmari was sediment in the Neo-Tethys Sea on the border of the Arabian Plate [12]. Its name is selected from Asmari Mountain in the southeast of Masjed Soleyman[1]. The thickness of Asmari differs between 200-1000 m.

Lithostratigraphic boundaries of the Asmari Formation in studied oil field are marked by shaly sediment of Pabdeh Formation at the base and evaporite sediment of the Gachsaran Formation on the top [13].

[1] A city in southwest Iran

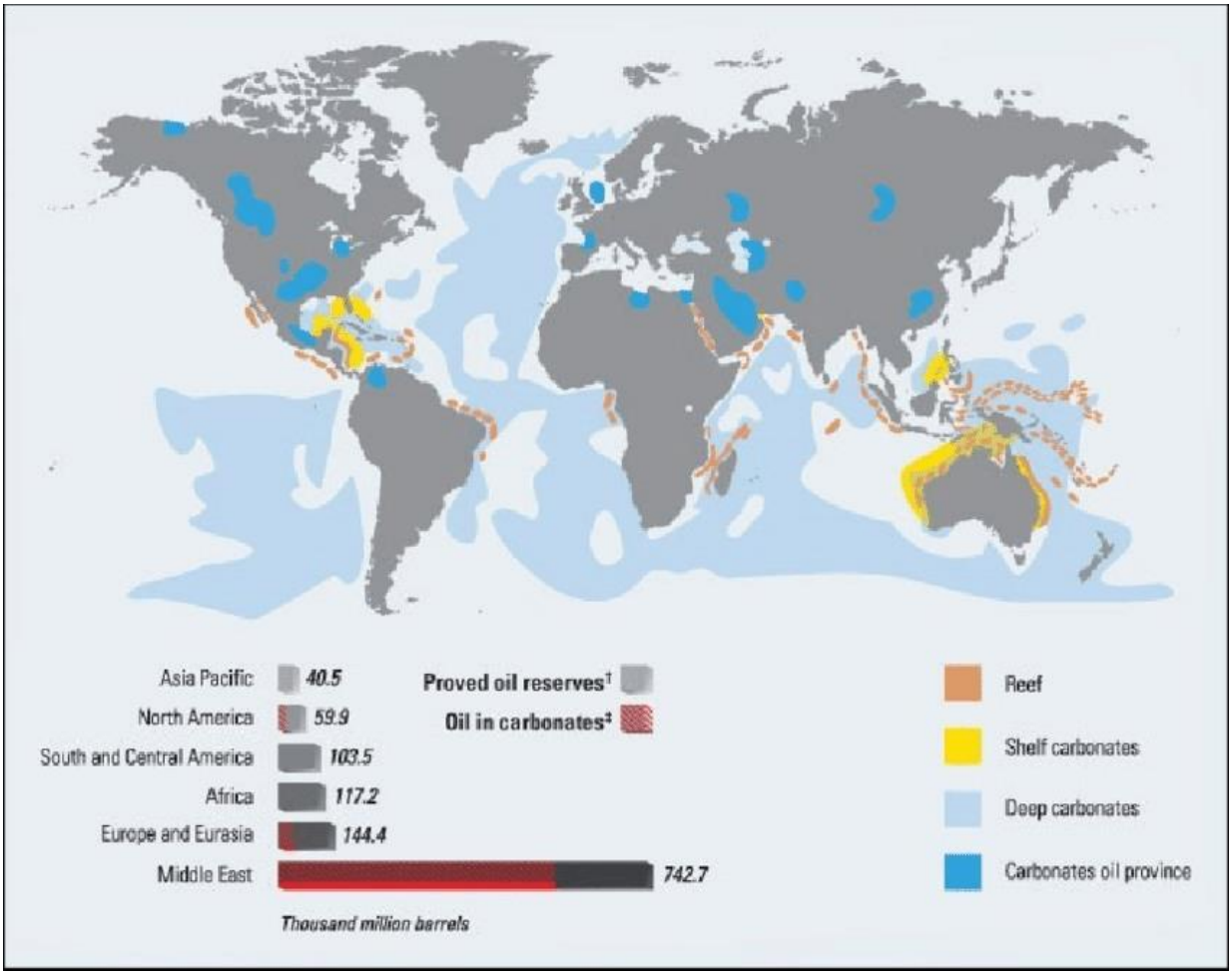


Figure 2.2 Location of different type of carbonate reservoirs in the world [1].

Tectonically, Asmari carbonate strata deposited retrogradationally shortly after a period of intensified late Eocene thrust faulting in the deformational wedge, and an overlying succession of upward-coarsening, northeasterly-derived siliciclastic deposits of lower Miocene to Recent age which is composed of erosional byproducts of the southwest-vergent Zagros thrust sheets [14]. Surprisingly, sandstone makes more than half of the lithological stratigraphy of Asmari formation in some of the Iranian oil fields.

Asmari is the main oil reservoir in more than 40 giant oil fields in Dezful embayment (Figure 2.3).

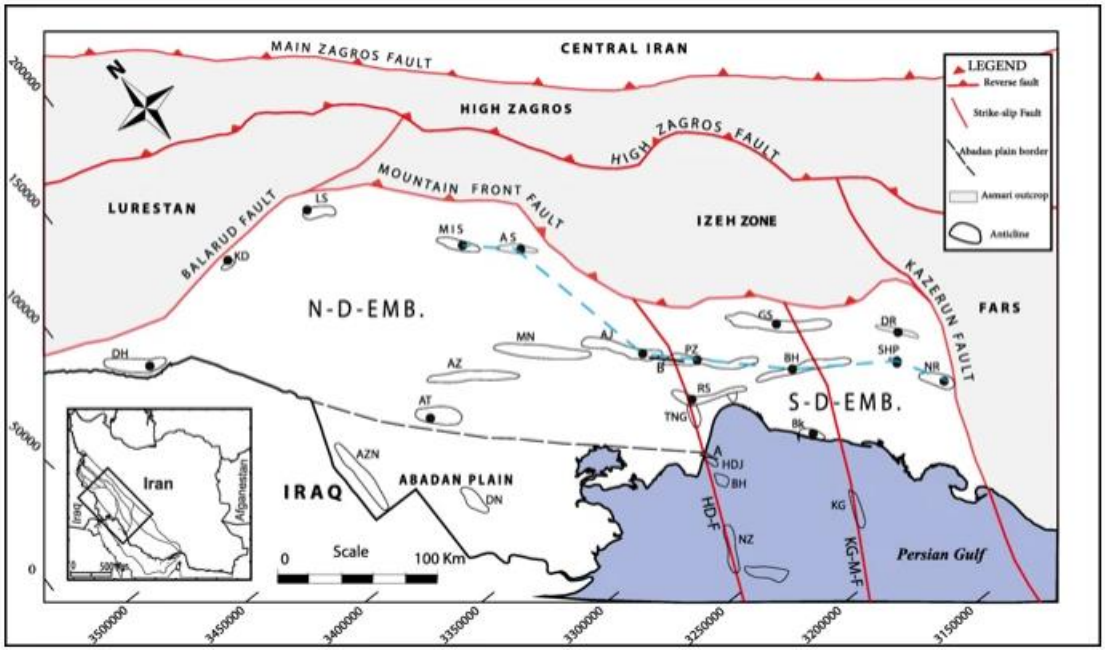


Figure 2.3 Dezful embayment and location of some giant oil fields inside [15].

High frequency of fracturing in Asmari's outcrops, high mud lust during drilling inside Asmari, pressure relation between wells with even more than 7 miles distance in individual oil fields, and finally the existence of high production zones in Asmari reservoirs are indicators that show Asmari is fractured [16, 17].

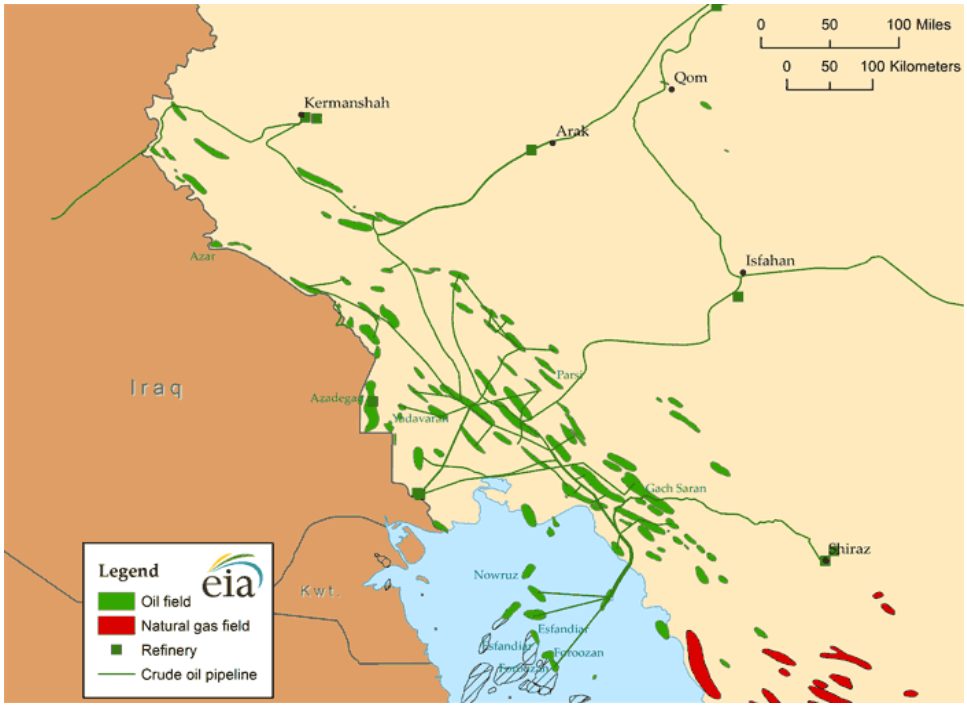
Asmari is classified into three sections: upper, middle, and lower. Techtronic, stratigraphy and petrology studies show that fractures, especially in upper and lower Asmari, are the main controlling factors of porosity and permeability. Diagenesis properties, dolomitization, and secondary dissolution are other controlling factors of porosity and permeability, especially in middle Asmari [14, 16].

2.3 Importance of Asmari Formation

Asmari in Iran is not only the most important oil reservoir (Figure 2.4.a) but also it is the most common host rock of dams (Figure 2.4.b).

Iran holds more than 361 billion barrels of oil equivalent in proved reserves of oil and gas, which positions the country as the top reserve holder in the world. This includes 9.3 percent and 17.2 percent of global oil and natural gas reserves respectively [18].

About 80 percent of Iran's oil is produced from Asmari[2]. The point is that the average porosity of Asmari is less than 10%, and oil mainly flows from fractures, which originated from tectonic tensions.



(a)

[2] Reports of Iran's National Oil Company

(b)

Figure 2.4 (**a**) Iran's largest oil fields in which Asmari is the reservoir of the majority of them. (**b**) Seimareh dam was built over Asmari.

2.4 Fracture Controllers

Faulting and folding (tectonic events), rock type, and thickness (lithological events), and porosity (diagenesis event) are the main factors which control the fracture generation. Those are briefly explained as follows:

2.4.1 Tectonically Structures

As aforementioned, tectonically structure (folding and faulting), are important factors in the estimation of fracturing. Fracturing potential in the upper layers of a folded formation, because of more bending situations, is more than the middle and lower layers (Figure 2.5). Therefore, Murray [19], Nemcok and Lisle [20], and McQuilan [21] have used this fact to estimate fracturing potential in folded formations. Dip of the surface is equal to the first derivative of the surface to depth or altitude, while the second derivative gives surface curvature.

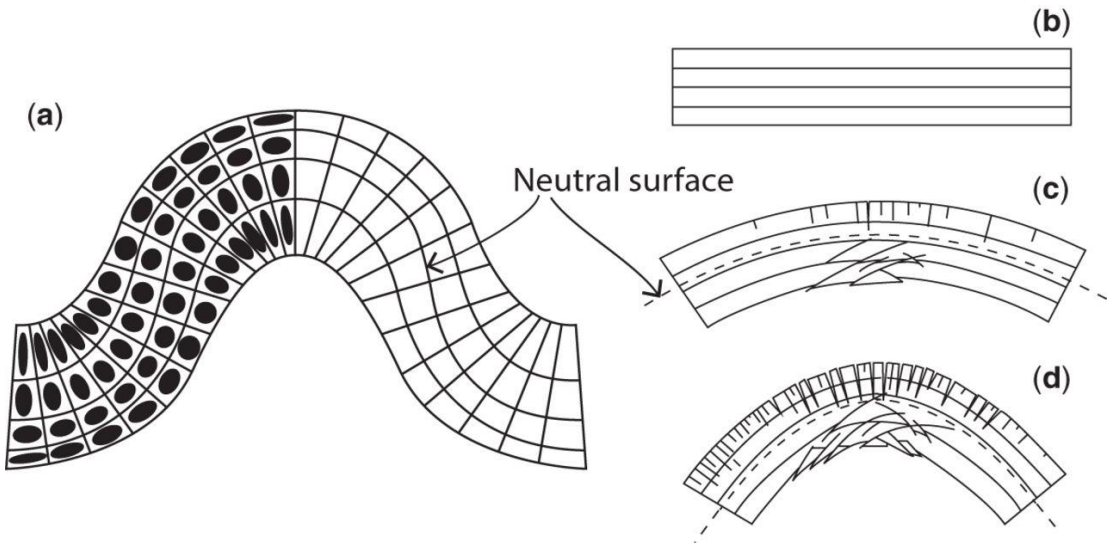
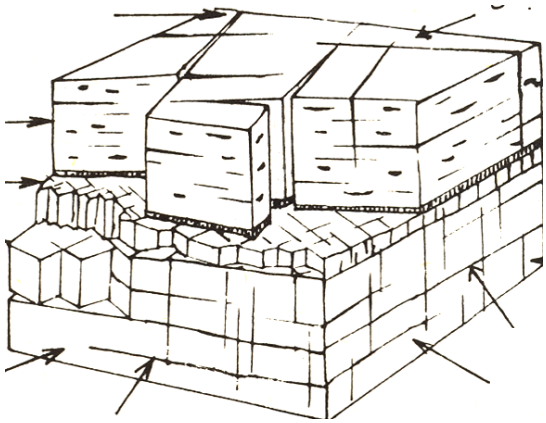


Figure 2.5 (**a**) The strain distribution within a buckle formed in a homogeneous, isotropic layer such as a massive, unbedded limestone or sandstone bed. (**b**) The undeformed layers prior to folding. (**c**, **d**) High strains are concentrated in the hinge region which is therefore the site of most intense fracture formation. From Ramsay [22].

Faults are the locations of high strains, in which high fracturing potential is anticipated. By approaching faults, the possibility of fracturing increases. By increasing the relocation of faults, fracture density as well as fracturing buffer zone increases. The fault's depth and genesis and the lithology are other well-known factors over fracture density [19, 20, 21].

2.4.2 Thickness

The possibility of fracturing increases by decreasing the thickness of layers (Figure 2.6). Some exceptions have been observed, which shows the importance of engineering interpretation.



(a)

(b)

Figure 2.6 (**a**) Increasing fracture density in low thickness layers. (**b**) Fractures in Asmari formation in Seimareh dam site.

2.4.3 Lithology

Mechanical resistance varies for different rocks. A certain stress may create failure in some, while deformation in other rocks. In the case of failure, fracture density will differ in different rocks. For example, fracture density in dolomite is much higher than limestone.

2.4.4 Porosity

Porosity is another dominant factor that controls fracturing. Porosity decreases rock strength and increases its tendency for deformation. Consequently, fracture density decreases by increasing the porosity.

2.5 Literature Review on Fractures in Asmari

Unfortunately, despite the importance of fractures in Asmari, limited related studies have been done, they will be briefly cited as follows.

2.5.1 Origin of Fractures

McQuillan's hypothesis was that fractures originated after sedimentation of Asmari and seismology shocks, therefore, he did not believe in tectonically theories [23]. Based on his theory, sediment surface morphology and physical characteristics of Asmari are controlling factors of orientation of fractures. He has mentioned that future faulting of fractured rocks has been the reason of propagation, especially open fractures. Colman-Sadd [24] and Rezaie and Nogole-Sadat [25] were famous researchers who have considered that tectonic is the origin of Asmari's fractures. Rezaee and Nogole_Sadat found a positive correlation between layer curvature and fracture density in Asmari formation in Rage-Sefid oil field; therefore, they addressed tectonism which has made curvature, as the genesis of fractures [25]. Based on their hypothesis, the collision of Arabian and Iran's plates, has controlled the folding of Asmari and consequently its fractures. It should be mentioned that for about half of the century, curvature is considered a controlling factor of fracturing in Iranian oil fields. Because fractured zones are oil production areas, experts utilize curvature to distinguish high production zones [26].

2.5.2 Different Kinds of Fractures in Asmari

Massive clusters of millimeter spaced cemented thin fractures that have crossed thick bedded limestones were observed in Asmari's outcrops [27]. It is considered a property of Asmari's fractures, because of the high frequency of similar observations; however, some researchers have doubted its depth of generalization. Theoretically, occurrence of thin fractures in the fold's crest could be anticipated [27].

Integration of the information of cores, well lost, drilling speed diagrams, well tests, FMS[3] , and similar logs confirm that the fracture's effect on fluid flow in Asmari, are located in crests. Aperture and length of those fractures are in the range of millimeter-centimeter and ten-hundred meters respectively.

Some studies that have been conducted in cores of Asmari in different oil fields, have shown an occurrence of frequent shear zones. Shear zone is an indicator of fault. Overall, drilling speed diagrams and pore pressure are indicators to show shear zones and faults. Watson could not find a strong relationship between those indicators and shear zones/faults in Asmari in Aghajari, Ahvaz, and Gachsaran oil fields [28]. He, of course, has reported multiple faults and shear zones in Asmari, with a length of more than a hundred meters. Therefore, fracturing and discontinuity in Asmari is a complex phenomenon.

Accumulative fractures are other well-known kinds of discontinuities in outcrops of Asmari. Their length differs from tens to hundreds of meters. Occurrence of these kinds of fractures in reservoirs is also possible. Their generation time affects their role on fluid flow. They are usually cemented if they were generated before oil migration; therefore, they act as a barrier in front of fluid flow. In the case of generation after oil migration, they could not be cemented, and could be a high

[3] Formation MicroScanner Images (FMS)

permeability zone in the reservoir. Some researchers have considered these kinds of fractures the main reason for hydraulic connection between Asmari and Bangestan[4].

Studies of cores and image logs show that the majority of discontinuities in Asmari are perpendicular to the layer's surface. This data also shows that open fractures parallel to the layer's surface in Asmari are rare.

It could be concluded that Asmari's fractures are in the following groups: thin fractures, accumulated fractures, feeder fractures, and shear zones. Also, the majority of fractures in Asmari are perpendicular to the surface of layers.

2.5.3 Relative Time of Fracturing in Asmari

Both open and cemented closed fractures were observed in Asmari. The time of generation of open fractures helps to distinguish that fractures were created before or after oil migration. Also, it helps to know when fractures were cemented.

McQuillan [29] believed that a wide variety in the generation of Asmari's fractures as well as the lack of relation between fracture density and folding, confirms that fractures were generated before the main phase of folding in Miocene. McCord & Associates [30] found that open fractures are located above the oil-water contact, and all fractures located lower than that contact were cemented. This is while oil migration from Bangestan to Asmari is evidence that shows that fractures were open before the migration. Halsey & Lapare (1978) [27] also believed that open fractures were generated after oil migration. Halsey & Corrigan [27] and Sepehr and Cosgrove [31] have interpreted fracture generation in two phases: first, the time compressional stresses were activated and consequently folding was commenced. Based on their theory, conjunction fractures

[4] Bangestan is another well-known reservoir in Iran, which is deeper than Asmari. Gurpi-Pabdeh, which is the source for Asmari and cap for Bangestan, is the Shale formation between those two reservoirs.

were generated in this phase. Phase two was commenced when curvature of folding caused tensile forces in upper Asmari, which created open fractures in the fold's crest. The fact is that their theory is compatible with the behavior of fractures in the upper, middle, and lower Asmari, in the majority of oil fields. It could be mentioned that in some of the oil fields, e.g. Khami, the behavior of fractures differs from their theory.

A group of researchers have focused on the geological history of the Arabian platform and its effect on North-South linear trend of fractures in Asmari [21, 24, 29, 32, 33, 34, 35, 36]. Some of which believe that a portion of fractures was generated because of the application of the host rock's faults. They reported that those faults were generated at late Proterozoic and early Cambrian, and reactivated in Triassic and late Cretaceous.

Gholipour [27] believed that anticline oil trap was generated before oil migration. Based on their theory, fractures were cemented because they initially were generated in a water-saturated environment in Miocene. In the second phase, Gholipour has reported that in some sections of anticlines fractures were opened; however, because of their extreme roughness, permeability was low. He believes that these kinds of fractures are geologically located in the middle of anticlines, where oil production of wells is not considerable. The second phase of plate movements happened in Pliocene, when anticlines were expanded. Generated fractures in this phase were outspread in planges, where fluid flow is high in oil fields. And finally, the third phase of plate tectonics which have continued so far, caused laminar fractures with aperture less than 0.1 mm which connect porous media by feeding fractures.

2.5.4 Fractures in Different Locations of Anticlines

The earlier theory was that high frequency zones are located in highly dipped plunges of the anticlines. Drilling rejected that theory. In order to develop a reliable model, aerial photography,

by scale 1:20000, of Asmari's outcrops in anticlines were done. That study showed the maximum fracture density could be observed in plunges. It also showed that linear discontinuities are highly propagated in axes of anticlines. It should be mentioned that those observations are not compatible with some of the oil fields. For instance, production in plunges of Masjed Soleiman oil field is much less than its other locations. Crichton [27] has investigated the relationship between production and the location of the wells and concluded that wells located on a crest have the highest production. His main hypothesis was that maximum and minimum fracture density could be located in the area with maximum and minimum curvature respectively. Afterward, studies showed that his hypothesis cannot be generalized. Twerehnbold & McQuillan prepared UGC map[5] of Asmari formation in Gachsaran and Kuh Khoviz using the data of 26 production wells. They investigated the correlation between fracture densities and anticline's properties, and found that the maximum fracture density is in high curvature zones [27]. McQuillan also found that fractures are a controlling factor for production in BiBi Hakime Oil field.

Positive correlation between fracture densities was confirmed by Sangree and his colleagues [37]. Sangree [34] and McCord & Associates [30] in individual studies found that fracture spacing depends on the curvature of anticline, e.g. higher curvature, lower fracture spacing.

Smellie was the first researcher who investigated the structure of anticline's limb (convex and concave) and its effect over fracture density. Smellie, Tornhenbold, Eshghi, and McCord *et al*., in individual studies over Bangestan, Aghajari, and Maroun oil fields, confirmed that fracture density in convex is significantly higher than concave [27, 30].

In conclusion, it seems that in Asmari formation in different oil fields, fracture density in plunges, hinge zones, axial planes, and convex limbs is higher than other locations. Meanwhile, there is a

[5] Underground Contour Map

positive relation between the layer's curvature and fracture density. Finally, fracture density increases by approaching faults.

2.5.5 Fractures and Thickness of Layers

McQuilan [20, 38] has investigated the relationship between thickness of layers and fracture densities in Asmari's outcrops in Asmari, Pabdeh-Gourpi and Pahn mountains. His studies showed an inverse logarithmic relationship between those parameters in Asmari's outcrops; consequently, fracture density decreases by increasing the thickness.

2.5.6 Cementing Minerals

Halsey & Lapare's studies over cores of BiBi Hakime Oil field have shown that calcite, anhydrite, gypsum and dolomite are main fillers of fractures [27]. NISOC[6]'s recent report also confirms their results.

2.5.7 Recent Studies

The development of dip measuring technology and then image logs, have helped the identification of fractures in oil wells [8]. Those logs were run in various wells; however, they could not solve the problem of fracture identification in oil wells, because: 1. for the majority of oil wells that were drilled before developing the technology of image logs, just petrophysical data are available, and 2. image logs are expensive, therefore oil companies prefer to run them in limited wells. Therefore, researchers, often utilized other databases for fracture identification. Let's emphasize two main reasons that fracture study is important in Asmari: 1. Asmari is a fractured reservoir, so fractures

[6] National Iranian South Oil Company

play a critical role in fluid flow, and fracture modelling to production optimization is necessary, and 2. fractures may cause instability in oil wells, so fracture identification in geomechanical studies is important.

Various projects and research were done for identification and modelling of fractures, some of which are inaccessible. Gholipour [39] has published some of the reports and maps that were a result of research done by a consortium. Based on those studies, fractures and faults which are parallel and perpendicular to anticline's axes, have the highest frequency and impact in oil production. Shaban has investigated the relationship between Asmari anticline curvature and fracture density in Gachsaran oil field [40]. He has found that curvature could be used for fracture modelling, which of course is not a novel result. It should be mentioned that in professional reservoir characterization software, e.g. Petrel or FracaFlow, curvature is used as a main input for fracture modelling. Nemati and Pezeshk [32] utilized core data to investigate the effect of petrology rock type, porosity and permeability on fracture density. They have found positive correlation between fracture density for dolomite and negative correlation with silica and anhydrite. They found that calcite, porosity and permeability don't affect fracture density. Wenberg and his colleagues found that layer thickness and texture are two main factors which control fracture density in Khoviz anticline [43]. They studied Asmari outcrop in the Asmari Mountain and by synthesising the origin of fractures have developed some ideas to generalize surface data to depth [44].

The relation between Asmari's anticlines and fracture density and their effect on production was the subject of the research of Stephenson *et al.* [45]. They found that main fracture zones were located in anticline's axes which are also the main fluid flow zone. Ahmad Hadi and associates have suggested two different times for fracture generation [43], based on a huge database of

Khoviz, Asmari, Solak, Deil, Razi, Sephid and Bangestan anticlines. The first set, before the folding of Zagros, concurrent with stress resulted in the collosion of Arabian and Iran's plates. They believed that a decrease in sedimentary surface caused fractures parallel to layering. The reactivation of faults in Mio-Pliocene addressed another factor of fracturing before the main phase of folding. Ahmad Hadi and associates considered folding the second reason of fracturing in Zagros.

2.5.8 Data Mining and Fracture Detection

Fracture detection by applying wavelet transform on petrophysical and well logs, has been one of the approaches that Nemati [46] and Mohebi *et al*. [47] utilized. They have addressed high frequency bands of petrophysical/well logs as indicators of fractures. Sonic and Gamma Ray were the logs that were used in their studies. Sahimi and Hashemi also applied wavelet transform over porosity log and addressed fractures by high frequency bands [44]. They have verified their proposed algorithm by a high variability of permeability log. Afterward, other researchers confirmed that their approach is not general and in an optimistic view it can just discriminate 10 percent of fractured zones from non-fractured ones [45].

Hsu and associates utilized sonic waves for the identification of open fracture, as well as finding their dip and hydraulic conductivity [48]. They have found that the intensity of decreasing the energy of low frequency band of Stoneley waves is a useful indicator.. It should be mentioned that they have confessed to the uncertainties of interpretation. Flavio and Gregor [49] proposed a composite well log, called velocity deviation log, to estimate different reservoir properties, and fracture identification as well. They estimated a sonic log by using a density log, and then calculated their proposed log by subtraction of estimated sonic log from common sonic log. The proposed log could be negative, zero or positive. Based on their study, wash out, fracture or high

gas content, are the reasons for negative values. By filtering wash out and gas content, which could be detected using other logs, fractured zones could be detected.

Daiguji *et al*., decomposed seismic data to fault cognition [50]. Later, it confirmed that their approach works; however, the problem is that its resolution is not suitable for fractured zone detection. Behrens *et al*., integrated seismic data for fracture detection, which was associated with high uncertainties [51]. Cores and PLs were utilized for discrimination of natural fractures from drilling induced ones during interpretation of image logs [52]. Martinez-Torres utilized fuzzy logic to integrate caliper, gamma ray, sonic, self-potential and resistivity logs to make a composite fracture log [4]. The lack of image log for verification of the proposed approach, is his research's shortcoming. Tran's dissertation was about characterization and modeling of naturally fractured reservoirs, but he has also tried to identify the fractures as a data source [53]. He also integrated different logs together, and then classified achieved log to discriminate fractured zones from non-fractured ones. Again, lack of an image log, is the reason that he could not confirm the validity of the proposed approach.

Dutta *et al*., proposed a new approach for the identification of open fractured zones [5]. They synthesized shear waves to identify fractured zones, and utilized image logs for verification of the results. Their approach has been especially acceptable for fault detection. Ozkaya and Siyabi tried to identify the open high-density fractured zones using dynamic properties of well [54]. They have ranked the parameters for fracture detection by factor analysis, and found that mud loss is the best. Yan and associates [55] proposed a fracture cased porosity modeling procedure. They created a synthetic image log by integrating PLs. They have found that the procedure is not only useful for fracture detection, but also for fracture properties estimation. It seems this procedure is straight-forward, and might be developed in the future.

Tokhmechi *et al*, decomposed the water saturation data by wavelets and showed that the majority of information of the original log is hidden at low frequency bands. They utilized gamma ray to filter zones for more reliable fractured zone detection. Approximated section of water saturation of filtered zones helped them to achieve accuracy of about 70% in discrimination of fractured zones [56]. They, in another study, applied Parzen and Bayesian classifiers to raw, de-noised and various frequency bands of logs to fractured zone detection [9]. They have confirmed that low-frequency bands of de-noised logs are the best data for their studies. They utilized OWA (ordered weighted averaging) to fuse the results obtained from different train wells and classifiers as well. The generalization of their approach was confirmed with an average accuracy of about 72%. Their third study was focused on fracture density estimation [57]. They utilized wavelet and Fourier transform to calculate energy of the PLs in the fractured zones. Linear and non-linear regressions were fitted between energy of logs and fracture densities. Results showed that there is strong correlation between the energy of caliper, sonic (DT), density (RHOB) and lithology (PEF) logs with fracture density in each well.

Jafari *et al.*, proposed an adaptive neuro-fuzzy inference system for fracture density estimation from PLs. They found a good statistical correlation between fracture density and well log data including sonic, deep resistivity, neutron porosity and bulk density, and achieved correlation coefficient of 98% between the measured and neuro-fuzzy estimated fracture density [58].

Mazaheri *et al.* developed Fracture Measure (FM), as a novel fractured zone detection criterion [59]. They defined FM as a parameter calculated by aperture, fracture type, azimuth and apparent distance. Artificial Neural Network was used to estimate FM and find the relation between FM and conventional logs, which resulted in satisfactory results with a generalized correct classification rate (CCR) equal to 80% [60]. Afterwards, in another study, they optimized cell size

in order to more effectively utilize FM to recognize fractured zones [59]. They applied different data mining techniques to do that, and fused the results with the Sugeno integral method. Comparison between the results confirmed that 30 cm is the optimum cell size [59].

Aghli *et al*., tried to find a quick generalized method for identification of fractured zones using PLs, and then used the Velocity Deviation Log (VDL) to identify fracture aperture opening and their effects on porosity and permeability. They found that the differentiation method could be easily used to recognize fractured zones in high fracture density zones by PLs [61].

Zarehparvar Ghoochaninejad *et al*., estimated hydraulic aperture of detected fractures using well log responses using a teaching–learning-based optimization algorithm (TLBO), which trained an initial Sugeno fuzzy inference system [62]. No matter how big the aperture size, they have proven that estimations are reliable in both conductive and resistive mud environments.

Mazhari and associates proposed a generalized case-based reasoning (CBR) method for fractured zone detection via PLs [10]. To such aim, they used a set of train wells to beget a database composed of both petrophysical data and the image logs. They conducted a learning automata-based algorithm to find the optimal similarity relation between PLs and manual interpretation of the borehole image logs. Their developed model was successfully tested on the Asmari reservoir through several oil wells and achieved a general CCR of about 70%.

The discrimination of lithofacies in order to optimize fractured zone detection using PLs was another approach to maximize the accuracy [63]. Mentioned chained activities shows the importance of using data mining methods to discriminate fractured zones from non-fractured ones, which is also the approach of current research.

Olivia *et al.*, developed a fast region-based Convolutional Neural Network (fast-RCNN) for automatically interpreting acoustic image logs to identify fractures and break outs. They achieved

around 81% accuracy in fracture detection which is satisfactory [64]. However, it should be mentioned that they just automatically interpreted image logs instead of manual interpretation. Therefore, their aim was not developing a method for discrimination of fractured zones by using PLs.

2.6 Conclusion and Thesis' Approach

Fractures have a main role in fluid flow and oil production; therefore, in fractured reservoirs, e.g. Asmari, they might be identified and modelled. Image logs (ILs), with a resolution of about 10 mm, are well-known tools for fracture identification, but are usually not available. Seismic sections, well test, mud lost, petrophysical logs (PLs), and core description that have been used for fracture identification [65, 66], were associated with serious shortcomings. Seismic sections have low resolution (10 meter even less), and could be used just for detection of faults or thick fractured zones. The resolution of well tests and mud lost is also low, meanwhile well tests are rarely available. Cores are usually unavailable, or not-oriented, therefore their interpretation suffers from various shortcomings. PLs are usually available, and their resolution is about 15 cm.

Recent research was focused on identification of fractures using PLs. Researchers have selected the wells, both PLs and ILs were run inside, in which fractured zones are known by interpretation of ILs. They have applied data mining/machine learning techniques to find the relationship between fractured/non-fractured zones and PLs. In abstract, a two class problem must solved. Based on my literature review, Parzen, Bayesian, and Cased-Based Reasoning were classifiers that were used for fracture zone detection.

In the current thesis, two modifications on previous approaches will be done. First: two classes are not balanced. Overall, non-fracture zones are more than 70% of the reservoir. The effect of

balancing between two classes on discrimination will be investigated. Second: other powerful classifiers are available, and they might help to better discriminate fracture zones from non-fracture ones. Random forest and deep learning are selected classifiers for this study. Outputs will be fused to achieve more accurate, reliable and general results, which in previous studies were used.

Methods will be applied on eight oil wells of a huge fractured reservoir. Their PLs and ILs are available.

3

Materials and Methodologies

3.1 Introduction

In the current chapter, image logs, PLs, and the effect of fractures over PLs will be introduced. Afterward, studied reservoir and database will be addressed. Preprocessing of the database is the title of the next section. Finally, the results of the studies, which is in fact the core of the thesis, will be presented.

3.2 Image Logs

Image logs image the wall of the wells by using electrical resistance or travel time of ultrasonic waves. Density based image logs were also recently developed. Bore Hole Tele Viewer (BHTV), are supplementary image logs, which have 360 degree rotation capability and take images of the wall of bore hole.

Electrical image logs record the micro resistance or micro conductivity of the walls [3, 67, 68, 69, 70]. This kind of image logs, which were called OBMI[7] or FMI[8], contain two, four or six pads with at least 16 electrodes laid over each pad (Figure 3.1). Therefore, they probe around the well

[7] Oil Based MicroImager
[8] Formation MicroImager

by resolution in the range of 2.5 millimeter. Their coverage also depends on the number of pads. For instance four pads cover 40% of the walls around the well (Figure 3.1). OBMI could record resistivity of less than 1 to more than 10000 ohm meter. The mentioned features help them to record slight variabilities, e.g. fractures.

Figure 3.1 Four pads Oil Based MicroImager

Image logs that were used could be run in water based conductive muds, or oil based nonconductive ones. FMS[9], FMI, ARI[10], EMI[11], and STAR[12] are examples of image logs that could be run in water based muds, were designed by different companies. OBMI and EI[13] are examples of those that could be run in nonconductive muds.

Overall, the resolution of image logs that could be run in nonconductive muds, are less than other classes. Also, ultrasonic based image logs are better than resistivity based, not only for fracture detection, but also for aperture calculation. The reason is that ultrasonic based image logs are not sensitive to the mud kind, and their horizontal resolution is so much higher than resistivity based.

[9] Formation Micro Scanner
[10] Azimuthal Resistivity Imager
[11] Electro Micro Imager
[12] S Tool A Resistivity
[13] Earth Imager

UBI[14], CBIL[15], and CAST[16] are examples of ultrasonic based tools that were frequently run for fracture detection [3, 71].

Some of the properties of image logs are abstracted in Table 3.1.

Table 3.1 Some of the properties of image logs.

Imaging Method	Mud Kind	Tools' Name	Resolution		Coverage Percent of the Wall	Number of Pads	Number of Electrodes in Each Pad	Logging Speed (m/hour)	Depth of Penetration of Electricity (mm)
			Horizontal (mm)	Vertical (mm)					
Electricity	Conductive	*FMS-4*	7.5	5	40	4	16	500	15-50
		FMI	5	5	80	8	24	565	15-50
		ARI	30 degree	200	100	---	12	1125	9
		EMI	5	5	80	6	25	584	750
		STAR	5	7.5	65	6	24	---	---
	Nonconductive	*OBMI*	30	30	32	4	10	1125	9
		Earth Imager	---	---	---	6	10	---	---
Ultrasonic	Conductive/Nonconductive	*UBI*	10	10	100	---	---	135	---
		CBIL	---	---	100	---	---	---	---
		CAST	1.8 degree	75	100	---	---	360	---
Borehole Imaging	Conductive/Nonconductive	*BHTV*	90 degree	11	10 percent	---	---	---	---

3.3 Database

The thesis database is related to one of the world's giant carbonate fractured reservoirs. Exploration of the studied field started in 1923, and the first oil well was drilled in 1931. Asmari, which is a fractured carbonate rock, is the formation of the reservoir. In the studied reservoir, the majority of oil is reserved in the rock's matrix, while fractures are the effective network for oil flow. More than 450 oil wells were drilled so far. Full set PLs were run inside almost all of the wells, while image logs were run in just eight of them. Therefore, developing a procedure to find the relationship between fractured zones and PLs in the wells containing image logs, and running it to find fractured zones in the wells that lack image logs, will help to increase the information

[14] Ultrasonic Borehole Imager
[15] Acoustic Borehole Image Log
[16] Acoustic Sonic Tool

about fractures by about 50 times. Obviously, it will help to achieve more reliable fracture models, geomechanical models and fluid flow simulation as well.

Petrophysical and image logs of eight wells were used in the current thesis. They are located over semi UGC[17] map of studied reservoir (Figure 3.2).

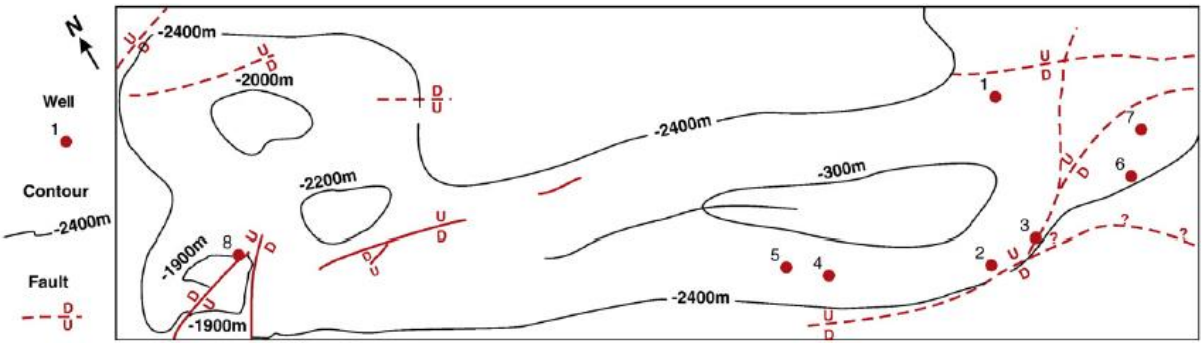


Figure 3.2 Semi UGC map and location of studied wells.

In Table 3.2 a list of PLs that were run inside eight studied wells are reported. Overall, 29 raw and interpreted logs were available in the studied wells. For instance, density (RHOB), sonic (DT) and photoelectric factor (PEF) are examples of raw and dolomite, porosity and summation of gamma ray (SGR) are examples of interpreted logs (Table 3.2). 16 out of 29 logs were selected for more studies. In Table 3.3, availability of selected logs in the eight studied wells are listed. Image logs were also run in studied wells. Interpreted image logs have been the source of discrimination between fractured zones and non-fractured ones.

Table 3.2 Discrimination of available PLs to raw and interpreted ones.

Kind of log	Logs
Raw	*Caliper, Gamma Ray (Potassium, Thorium, Uranium, GR), Resistivity (MSFL, LLS, SFL, ILM, ILD, LLD, DFL, RXO, RT), Sonic (DT), Density (RHOB, PEF), Neutron (NPHI)*
Interpreted	*Lithology (Limestone, Dolomite, Sandstone, Shale, Anhydrite), Gamma Ray (SGR, CGR), Porosity and Saturation (POR, PHI, TOTAL, SEC, 2ASW ×PHIO, SXO, SW)*

[17] Underground Contour map

Table 3.3 Availability of PLs in studied wells

Well / Well Log		1	2	3	4	5	6	7	8
Caliper		*	*	*	*	*	*	*	*
Geological Rock Type	Limestone	*				*	*	*	*
	Dolomite	*			*	*		*	*
	Sandstone				*	*	*	*	*
	Shale					*		*	*
	Anhydrite	*			*	*		*	*
Gamma Ray	Potassium	*	*	*		*			*
	Thorium	*	*	*		*			*
	Uranium	*	*	*		*			*
	CGR	*	*	*	*	*	*	*	*
	GR	*			*				
	SGR	*	*	*		*	*	*	*
Resistivity	MSFL				*				
	LLS(SFL)				*				*
	ILM		*				*	*	*
	ILD (LLD)	*	*	*	*	*	*	*	*
	DFL						*		
	RXO				*	*		*	*
	RT	*			*	*	*	*	
Sonic	DT	*	*	*	*	*	*	*	*
Density	RHOB	*	*	*	*	*	*	*	*
	PEF	*	*	*	*	*	*	*	*
Neutron	NPHI	*	*	*		*	*	*	*
Porosity and Saturation	Porosity (PHI)	*			*	*	*		
	Total					*	*		
	SEC				*	*	*		
	2ASW*PHIO				*		*	*	*
	SXO				*			*	*
	SW	*			*		*	*	*

* means available

As can be seen in Table 3.3, some of the logs are not available in different wells. For instance, geological rock type logs, e.g. dolomite, shale or limestone are not available in about half of the wells. Raw PLs containing caliper, cumulative gamma ray (CGR), SGR, RHOB, DT, PEF, and neutron porosity (NPHI), are available in about all of the wells. Water saturation (SW) and Porosity (effective "PORE" and total "PORT") logs are also available in the majority of the wells. Based on the literature, they are also useful in fractured zone detection; therefore, it will be investigated how useful they are in this study. Resistivity logs containing microspherically focused log (MSFL),

shallow laterolog (LLS), spherically focused log (SFL), induction deep resistivity (ILM), induction log (ILD), deep laterolog (LLD), focused resistivity log (DFL), knowledge of resistivity of the invaded zone (RXO), resistivity (RT) are not available in a majority of the wells; therefore, they will not be used in the current study.

3.4 Preprocessing

Five following amendments have been done over the PLs:

3.4.1 Depth Shifting

Comparison between spikes in different logs shows limited depth mismatch. Depth shifting was done to make depth match between all logs.

3.4.2 Tools Pickup

Series connection of logging tools helps in decreasing the time and cost of running PLs. This makes up for the problem of the lack of data for some of the logs in the bottom of the wells, which is called tools pickup. The known code for these depths is -999.25. In this study, those depths were removed.

3.4.3 Tools Malfunction

Tools malfunction is the lack of data for some PLs in the top of the wells. Similar to tools pickup, depths corresponding with these data were removed.

3.4.4 Cycle Skipping

Rarely, petrophysical tools recorded a skipping, which were modified by replacing the mean of neighbor amounts.

3.4.5 Wash Out

In the case of washing around the wall of wells, no matter what its reason, the majority of logs will record mistake amounts. In the current study, the average of caliper is considered as bit size, and the difference between it and caliper is considered a wash out. The zones that differ more than 1.5 inch were considered wash out zones, and were verified by image logs. To prevent systematic errors, recognized zones were removed.

3.5 Making Fracture Log

As aforementioned, image logs and their interpretation of all eight studied wells are available. In Figure 3.3, image log of well 2 in the depth ranges from 2525 to 2550 meter is shown. Because of low resolution of PLs (Figure 3.4), it was not anticipated that they would be effective for discrimination of individual fractures. Therefore, image logs were used to discriminate fractured zones from non-fractured ones. A column, addressed by fracture added to the database, in which fractured zones were labeled by one, or else by zero. For example, in Figure 3.3, fractures were reported in the following depth, which is labeled by 1: 2528, 2531-2534, and 2537-2550 meters. In Table 3.4, an example of PLs of well 2 (depth: 2527-2533 meter) after adding fractured zones are displayed.

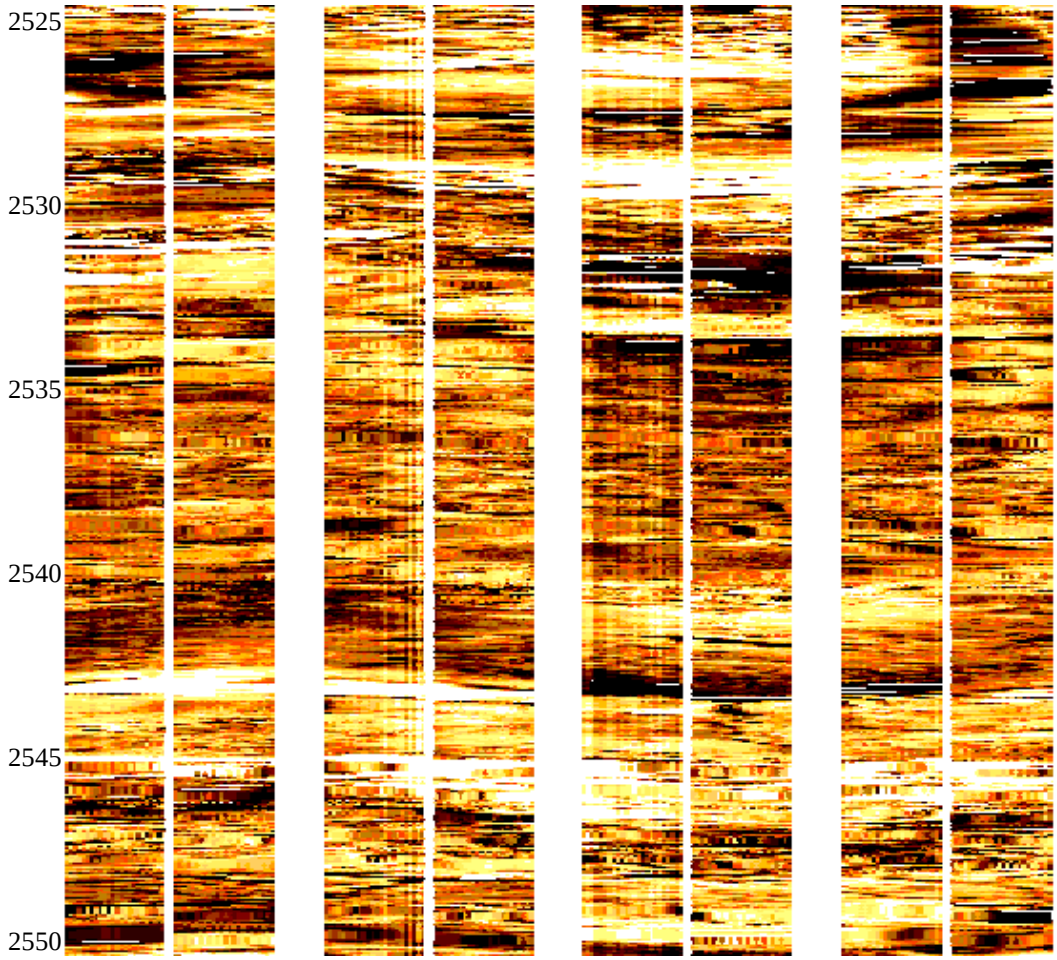


Figure 3.3 Image log of well 2 from the depth 2525 to 2550 meter.

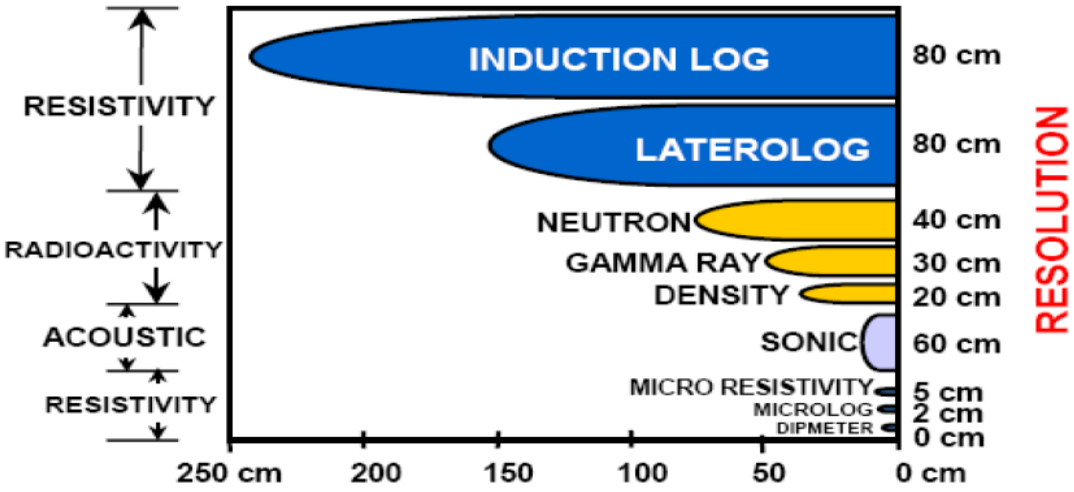


Figure 3.4 Resolution and leverage of different PLs.

Table 3.4 An example of PLs, fracture label was added to them in well 2.

DEPTH	Fracture	CALI	CGR	RHOB	SGR	DT	PEF	NPHI	POTA	URAN	THOR
2527.554	0	9.305	7.609	2.802	44.69	53.47	3.099	5.675	0.0043	4.261	0.5631
2527.706	0	9.432	6.79	2.809	45.9	53.65	3.103	5.65	0.0045	4.312	0.6458
2527.859	0	9.471	5.789	2.818	37.4	53.12	3.109	5.398	0.0005	4.163	0.5768
2528.011	0	9.507	5.071	2.827	34.39	51.99	3.121	4.782	0.0001	4.077	0.2779
2528.163	0	9.511	5.008	2.836	45.09	50.33	3.141	3.925	0.002	4.929	0.2908
2528.316	1	9.531	5.775	2.843	41.65	48.59	3.167	3.008	0.0011	4.831	0.047
2528.468	1	9.59	7.131	2.846	51.25	47.25	3.198	2.249	0.0028	5.245	0.824
2528.621	1	9.471	8.596	2.846	53.42	46.37	3.234	1.72	0.0032	5.308	1.014
2528.773	1	9.329	9.678	2.843	55.63	45.97	3.273	1.437	0.0037	5.37	1.249
2528.925	0	9.281	10.05	2.838	56.83	46.03	3.315	1.326	0.0037	5.392	1.471
2529.078	0	9.249	9.756	2.834	52.77	46.44	3.356	1.326	0.003	5.177	1.195
2529.23	0	9.234	8.908	2.831	46.03	47.09	3.387	1.386	0.0041	4.58	0.3413
2529.383	0	9.293	7.695	2.831	45.33	47.77	3.397	1.486	0.0045	4.456	0.2603
2529.535	0	9.234	6.426	2.831	38.33	48.3	3.38	1.612	0.0038	3.996	0.2448
2529.688	0	9.186	5.329	2.83	36.43	48.57	3.336	1.776	0.0035	3.904	0.3947
2529.84	0	9.198	4.55	2.826	38.59	48.6	3.272	1.954	0.0041	3.967	0.2406
2529.992	0	9.194	4.065	2.821	29.17	48.64	3.202	2.147	0.0004	3.66	0.0487
2530.145	0	9.202	3.909	2.816	35.28	48.84	3.145	2.328	0.0008	3.696	0.864
2530.297	0	9.174	3.927	2.813	36.02	49.26	3.121	2.497	0.0007	3.709	1.035
2530.45	0	9.15	4.044	2.816	33.72	49.78	3.14	2.619	0.0001	3.653	0.8623
2530.602	0	9.139	4.154	2.823	33.23	50.21	3.197	2.68	0.0001	3.638	0.7688
2530.754	0	9.123	4.528	2.832	34.41	50.46	3.275	2.661	0.0005	3.658	0.8446
2530.907	0	9.139	5.142	2.842	32.84	50.56	3.355	2.598	0.0007	3.559	0.5838
2531.059	0	9.139	6.265	2.85	35.4	50.63	3.418	2.51	0.0016	3.598	0.748
2531.211	0	9.131	7.768	2.854	32.62	50.71	3.453	2.431	0.0013	3.455	0.465
2531.364	1	9.15	9.374	2.856	43.37	50.75	3.458	2.354	0.0032	3.83	1.516
2531.516	1	9.139	10.7	2.856	43.12	50.69	3.44	2.264	0.0034	3.772	1.482
2531.669	1	9.15	11.59	2.855	45.5	50.47	3.411	2.135	0.0039	3.855	1.71
2531.821	1	9.154	11.94	2.854	44.13	50.17	3.383	1.979	0.0035	3.773	1.699
2531.973	1	9.162	11.8	2.85	40.31	49.83	3.36	1.836	0.0026	3.607	1.488
2532.126	1	9.15	11.28	2.843	39.62	49.54	3.338	1.76	0.0025	3.56	1.438
2532.278	1	9.139	10.72	2.833	41.91	49.45	3.312	1.772	0.0033	3.624	1.528
2532.431	1	9.139	10.24	2.822	40.62	49.58	3.278	1.868	0.0034	3.552	1.33
2532.583	1	9.139	10.03	2.81	33.51	49.96	3.231	2	0.0023	3.193	0.7956
2532.735	1	9.139	10.21	2.793	33.94	50.72	3.167	2.144	0.0027	3.196	0.7417
2532.888	1	9.139	10.73	2.768	42.09	51.97	3.086	2.427	0.0042	3.769	0.9276
2533.04	1	9.15	11.48	2.734	49.37	53.73	2.993	3.026	0.0064	4.644	0.0603
2533.193	1	9.162	12.13	2.694	52.97	55.91	2.907	4.194	0.0071	4.801	0.3093
2533.345	1	9.147	12.31	2.66	53.78	57.83	2.846	5.945	0.0073	4.862	0.3046
2533.497	1	9.257	11.9	2.643	62.21	56.84	2.825	8.049	0.0057	5.968	0.7976

3.6 The Effect of Fractured Zones on Petrophysical Logs

In the following, the effect of fractured zones over PLs are briefly explained:

Caliper:

Caliper tool reads the diameters of borehole very precisely when in fractured rock edge chipped away while drilling. Also, sometimes because of mud accumulation in open fractures the resulting caliper log is even less than bit size.

Gamma Ray (GR):

In Table 3.5, natural radioactivity of different rock types is reported. GR is natural radioactivity that comes from Uranium, Thorium, and Potassium that have accumulated in fractures. In the study wells, GR shows a sudden increase with any pick of Uranium. This is common in all the study wells.

Table 3.5 Natural radioactivity in different rock types [3]

Rock Type	API
Limestone	5-10
Dolomite	10-20
Sandstone	10-30
Siltstone	30-80
Shale	80-140

Uranium:

Uranium can dissolve in water and deposit in fractures and causes high pick in its log.

Thorium:

Claystone (shale) contains Thorium. Shale has plasticity properties and it is contrary to the properties of fractured rock. As a result, it is anticipated that Thorium increases in non-fractured zones.

Potassium:

Potassium is found in Feldspar and clay minerals. Because of the lake of Feldspar in the study formation, Potassium is due to clay formation. As Clay has plasticity, it contains fewer fractures. It can be inferred that an increase in Potassium is related to non-fractured zones.

Sonic (DT):

The sonic log is based on wave arrival time. This time depends on P wave velocity that passes through lithology, porosity, and fluids. If the wave path passes a fracture zone, the time will increase because P wave velocity is slower in porous and fluids rather than in solids.

Resistivity (RT):

By trapping brine in fractures, resistivity decreases. It means the non-fractured zones have a high RT.

Photoelectric Factor (PEF):

Fractured zones in some of the study wells indicate high PEF because of the tarp of water. It is because elements of connate water have a higher atomic number than rocks.

Density (RHOB):

Usually, because of the low density in the fractured zone toward rocks, RHOB is lower in the fractured area.

Neutron Porosity (NPHI):

NPHI would increase if a fracture zone filled with fluid or Hydrogen content.

Geological Rock Types:

Dolomite and limestone are brittle, with elastic behavior, capable of fracturing. Shale and anhydrite are ductile, capable of plastic behavior.

Water Saturation (SW):

There are some reasons water saturation could be affected by fractures:

- In water wet reservoirs, water content increases by increasing the contact surface between rock and fluid. Rocks of the current reservoir are water wet; therefore, SW might be an indicator for open fractures.

- In the case of using water-based muds (similar to the studied reservoir), a portion of water infiltrates to the open fractures and is replaced by the oil. Studies have shown that maximum replacement requires two days' time [72]. Well service companies, usually run the petrophysical tools with more than a two day delay. Consequently, higher water saturation is anticipated in open fractured zones.
- Water is usually located in upper parts of the reservoir. Fracture density in the upper Asmari is more than the middle and lower Asmari. Therefore, water saturation in fractured zones might be high, and an indicator for fractures. It should be mentioned that sometimes water saturation is high because of high shale content. Shale or CGR are the logs that could help to filter these zones.

3.7 Proposed Procedure for Fractured Zone Detection

Geological rock types, oil or water saturation, and porosity are the common properties that were captured from petrophysical well logs. In other words, PLs are not usually affected by fractures. For example, by increasing the clay content, resistivity decreases, and gamma ray increases, while fracture does not mainly affect those properties. Due to lack of a significant effect of fractures on PLs, researchers proposed data mining procedures for fractured zone detection from PLs. In the current study a two-step machine is designed to do that:

1. Designing a classifier in order to discriminate fractured zones from non-fractured ones by using PLs. Decision tree, random forest, support vector machine and deep learning, are classifiers that were used. It should be mentioned that those classifiers were not used in previous similar studies.

   Those classifiers were applied over PLs which are not balanced, and balanced PLs.

2. Ordered weighted averaging data fusion were applied on the outputs of classifiers in order to integrate results to achieve a more reliable-general discrimination between fractured zones and non-fractured ones.

Mentioned methods are described as follows.

3.8 Decision Tree

Decision Trees are like an upside-down tree that can perform both classification and regression tasks. The differences between Decision Trees and other classical machine learning models are that in Decision Trees users don’t need to perform any sort of data preprocessing such as handling the missing values and outliers and addressing the multicollinearity. Also, Decision Trees can handle nonlinear data and make a decision while the classical linear model couldn't.

Overall, construction of Decision Trees is like a tree that has a root node and several nodes named internal nodes or decision nodes. Internal nodes include a series of questions related to the data in order to reach a leaf node or terminal nodes that are decision nodes.

There are several steps to reach a final decision in leafs nodes:

Step1: Split data into multiple subsets according to the rule associated with the variable, in the root node.

Step 2: Split data again in internal nodes or decision nodes recursively according to the best decision to identify the variables and the rule associated with the variable.

Step 3: Repeat step 2 on the sub-nodes until reaching a stopping condition.

Step 4: Leaf nodes are the final decisions based on the majority class label for classification goals or consider the average of the target variables which presents a regression task.

Uusually, the tree-building algorithm is the same as mentioned above; however, there are some different tree-building algorithms like CART, CHAID, ID3, C4.4, C5.0, etc. In each of them, the criteria that are used for decision making might be different. For example, in the CART algorithm by Gini Index impurity selecting the best feature is present. Also in the ID3 algorithm, the information gains method, and for C4.5 algorithm, the Gain Ratio method is assessing the best data selection for the next nodes [73].

For the response to the question about how the best features to effectively split each node could be found, first the meaning of the homogeneity should be clarified. Homogeneity in a node means that the class label associated with the node belongs to a single class.

The best splitting node in Decision Tree classification refers to obtaining as homogenous as possible sub-nodes or child nodes upon splitting a parent node. This means the lower variance in sub-nodes is the better splitter in the case of regression. Classification Error, Gini Index, and Entropy are exemplar approaches that were proposed to achieve this goal. Since these approaches show the impurity of a node, a lower value of them means the higher the homogeneity of the node [73]. Let's look at these approaches in detail:

- **Classification Error**

Classification Error is the majority of probabilities associated with each node while equation 1 is correct [73]:

$$E = 1 - \max(P_i) \qquad (1\text{-}3)$$

where P_i is the probability of data point belonging to the i^{th} layer of classification or class label i.

- **Gini Index**

Gini is an index of the amount of random data points being misclassified. This index varies between 0 and 0.5. The lower Gini Index shows the lesser chances of any random data point that is misclassified and it helps with better decision making with lower ambiguity [73].

$$G = \sum_{i=1}^{k} p_i(1 - p_i) = 1 - \sum_{i=1}^{k} p_i^2 \qquad (2\text{-}3)$$

In which p_i corresponds to the probability of the data point belonging to i^{th} class label and k accounts for different class labels.

- **Entropy**

Entropy is an index of the degree of disorder present in data at a node. The range of this index changes between 0 or minimum disordering and 1 or maximum disordering rate in the class labels of the target variables [73].

$$D = -\sum_{i=1}^{k} p_i . \log_2(p_i) \qquad (3\text{-}3)$$

where p_i corresponds to the probability of the data point belonging to i^{th} class label and k accounts for different class labels.

There are some advantages and disadvantages associated to Decision Trees [73, 74]:

- **Advantages of Decision Trees:**
    - Decision Trees can be used in both classification and regression problems.
    - Decision Trees are a fast method for defining hyper parameters.
    - A decision tree doesn't need too much data preprocessing such as scaling or outlier treatment methods.

- A decision tree is not a complex mathematical method, it is easy to understand and interpret.
- It can be used to classify non-linearly separable data

- **Disadvantages of Decision Trees**:
  - Decision Trees couldn't present a linear relationship between the predictor variable and the target variable.
  - The subset of the numerical variable will be a single prediction value.
  - The inherited problem with Decision trees can lead to overfitting issues.

3.9 Random Forest

Random forest is a tree-based classifier that could classify with high accuracy, stability, and ease of interpretation. It classifies by using means of features or mode of categorical features. Random forest, which is a popular classification/regression algorithm in scikit-learn, is combining several decision trees that were created by different sets of observations. A final prediction is made by averaging the prediction of each tree. This is the benefit of random forest rather than decision tree, because it solves overfitting to the training data. This also leads to a higher accuracy in comparison with decision tree. Random forest could also be useful in ranking the features. Boruta is a random forest based algorithm used for feature selection [73, 74].

3.9.1 How Random Forest Works?

Four steps of the random forest are as follows [73, 74] (answer Figure 3.5):

- Select the samples from the dataset,
- Create a decision tree and predict the results for each selected sample,
- Vote all predicted results individually, by using the dataset mode factor,

- Find the final result, by using majority voting over predicted results.

3.10 Multilayer Perceptron (MLP) or Multi-Layer Neural Networks (MLNN)

The perceptron is a simple binary classification that comes from how the future looked from the perspective of the 1950s. For example, it predicts whether input belongs to a certain category of interest or not: fraud or not_fraud, cat or not_cat, etc. [75].

A multilayer perception (MLP) is a deep, artificial neural network and is an entrance gate to the deep learning world. The structure of MLP consists of an input layer to receive a raw dataset and the last layer that is an output layer that makes a final decision. Between these two layers (input/output layers), there is an arbitrary number of hidden layers that are the core of the MLP and consist of functions and weighting coefficients.

Multilayer perceptions train on pairs of input and output layers to learn the best model between those. Simultaneously, in training model functions, weighting coefficients, and biases are optimized in order to minimize the error between result data in the output layer and the test data.

Backpropagation is used to minimize the error. Error can also be measured in a variety of ways, for instance, root mean squared error (RMSE) [75].

3.10.1 Definition of Perceptron

A perceptron is a simple binary classification algorithm that divided the dataset into two parts “0” and “1”. It is modeled like a unit of the human brain, “the neuron”, to learn and solve complex problems. Figure 3.6 shows the function of the brain that depends on activities of neurons (brain cells) and synapses (connections) that receive and send electrochemical signals (messages).

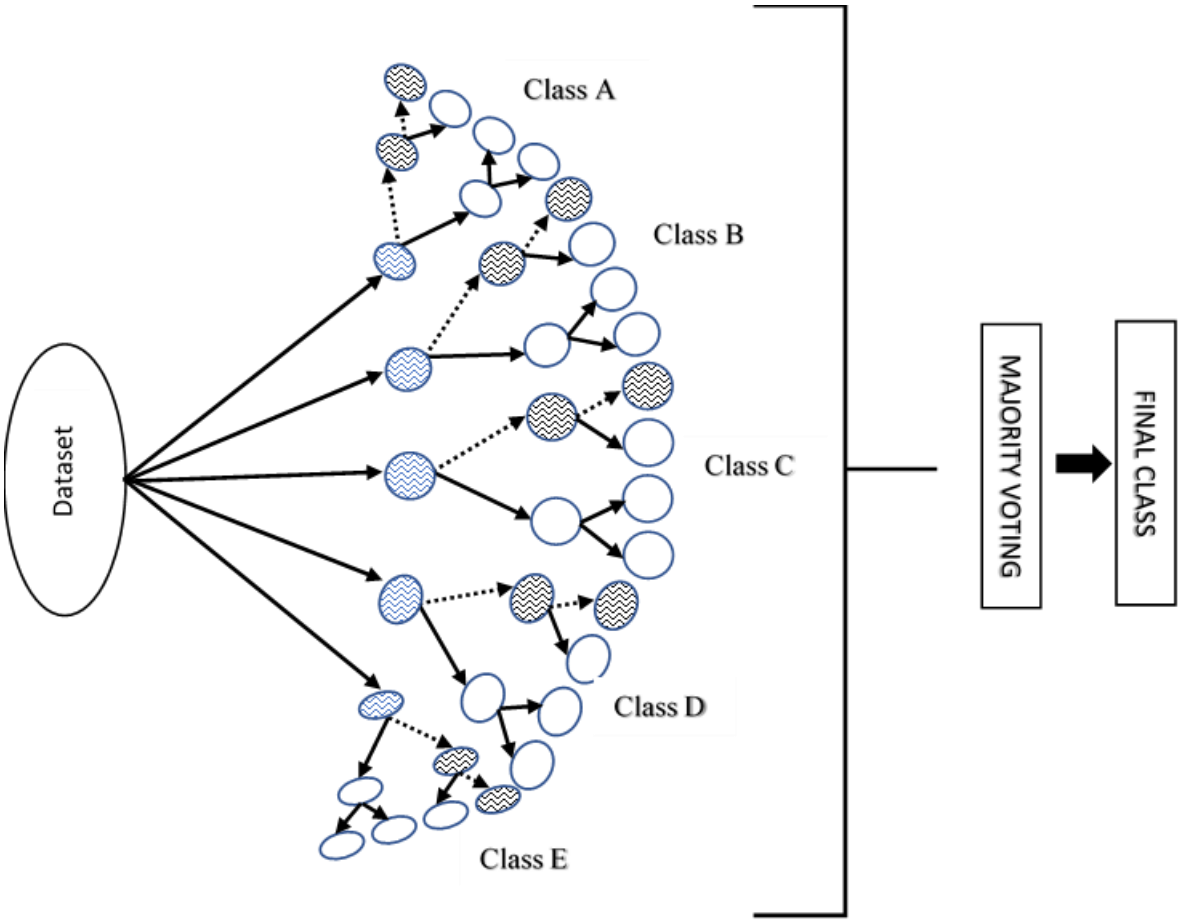


Figure 3.5 Random Forest Classifier

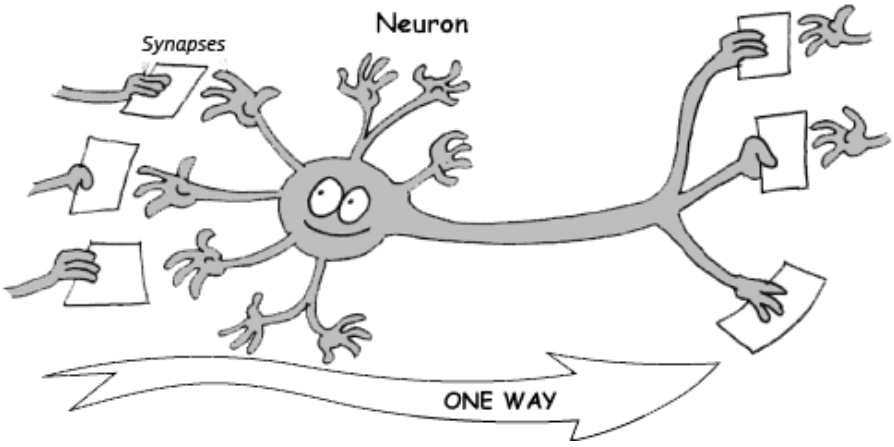


Figure 3.6 A conceptual procedure of perceptron [76].

A perceptron is a very simple machine learning model. It has some inputs that are a construct of weights to signify how important they are, and generate output decisions. By combining several perceptron's, an artificial neural network will be made. A neural network has the ability to solve a problem through enough training data and computing power [75].

3.10.2 Multilayer Perceptron

By stacking several layers in each there are several perceptron's, a multilayer perceptron will be made. Each layer of perceptron analyses the information and sends out the result to the next layer. Therefore the input layer, middle layers, or hidden layers, and the final layer or the output layer are the base of a multi-layer perceptron, often abbreviated as MLP (Figure 3.7).

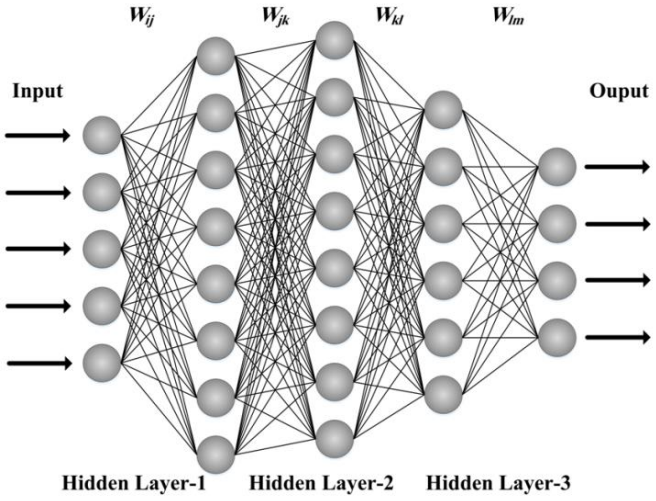


Figure 3.7 Multi-layer perceptron neural network (MPNN) [78].

The first assumption is a matrix $\in \mathbb{R}^{n \times d}$, as an input dataset by n sample where each has d features. The second assumption is that MLP has h hidden layers that are $H \in \mathbb{R}^{n \times d}$. The third assumption is the hidden layer's weights $W^{(1)} \in \mathbb{R}^{d \times h}$ and biases $b^{(1)} \in \mathbb{R}^{1 \times h}$ and the fourth assumption is the output layer $W^{(2)} \in \mathbb{R}^{d \times q}$ and biases $b^{(2)} \in \mathbb{R}^{1 \times q}$. Therefore, the outputs $O \in \mathbb{R}^{n \times d}$ of the one hidden layer MLP as follows [76]:

$$H = XW^{(1)} + b^{(1)},$$
$$O = HW^{(2)} + b^{(2)}. \quad (4\text{-}3)$$

A nonlinear activation function σ to be applied to each hidden node that outputs of activation functions are called activations. So, multilayer architecture will be as follows [76]:

$$H = \sigma(XW^{(1)} + b^{(1)}),$$
$$O = HW^{(2)} + b^{(2)}. \quad (5\text{-}3)$$

By stacking hidden layers, the next hidden layer will define one another as follows [76]:

$$H^{(1)} = \sigma_1(XW^{(1)} + b^{(1)}),$$
$$H^{(2)} = \sigma_2(H^{(1)}W^{(2)} + b^{(2)}). \quad (6\text{-}3)$$

The most popular activated functions are the rectified linear unit (ReLU), sigmoid function or squashing function, and the tanh (hyperbolic tangent) function.

3.11 Support Vector Machine

Support vector machines (SVMs) are one of the most powerful and flexible methods in machine learning and can be used for both classification and regression; however, it is recommended to be utilized as a classifier. Since SVM can handle multiple continuous and categorical variables, it has become a popular method. SVM classifies datasets by developing hyperplanes in multidimensional space. SVM could iteratively generate best-general hyperplane that contains minimum error. At the end of classification by SVM, hyperplanes with maximum margin (MMH) will be achieved to discriminate different classes from each other [79, 80] (Figure 3.8).

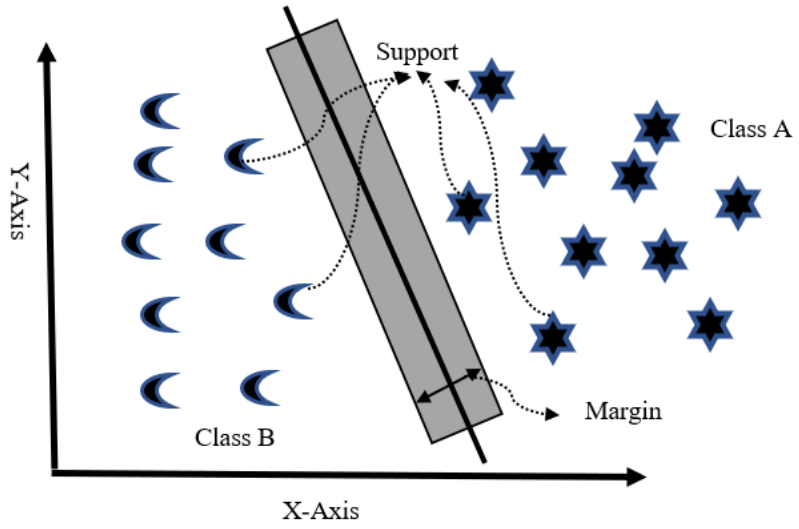


Figure 3.8 Support Vector Machine

The important concepts in Figure 3.8 are:

- *Support vectors*: Data points that are nearest to the hyperplane are called support vectors. The best separator line will be determined by these data points.
- *Hyperplane*: The separator or final decision plane which divides the dataset to the final class is the hyperplane.
- *Margin:* The distance between two lines that are closest to the support vectors in all of the classes are defined as margin. The bigger margin shows a better, more general classifier.

To find a better classifier by SVM, one should find the maximum marginal hyperplane (MMH). Therefore, SVM will generate several hyperplanes iteratively for the best discrimination and this method finds the best hyperplane that works correctly.

- **SVM Kernels**

Kernels in the SVM classifier convert a non-dimensional dataset to the separable space by adding more dimensions to it. Briefly, the SVM classifier is implemented with a kernel that transforms input data into the higher dimensional space with the best class discrimination.

The more popular kernels used by SVM are Linear, Polynomial, and Radial Basis Function (RBF) Kernels.

The linear kernel is the sum of the multiplication of each pair of input data and can be defined as a dot product between two vectors X and X_i. Its formula is as follows [79]:

$$K(x, x_i) = sum(x * x_i) \quad (7\text{-}3)$$

The polynomial kernel is defined for nonlinear input space. This kernel is generalized of linear format and is suitable for distinguishing curve shape space separation. Following is the formula for the polynomial kernel [79]:

$$k(x, x_i) = 1 + sum(x * x_i)^d \quad (8\text{-}3)$$

where d is the degree of the polynomial, which could be calculated manually in the learning algorithm.

The RBF kernel is a kind of exponential kernel that is common in SVM classification and maps input space in definite dimensional space. The following formula explains it mathematically [79]:

$$K(x, x_i) = \exp(-gamma * sum(x - x_i)^2) \quad (9\text{-}3)$$

in which gamma is in the range of 0 to1.

SVM classifier is an effective method when the number of features is quite large. This method works even if input data is non-linear. It is a powerful classifier model since it maximizes margin.

On the other hand, choosing the best fit kernel is a big issue as the wrong choice of kernel can lead to higher error. SVMs, although are a popular method, might need a big runtime, especially in big data problems. Also, they need an extensive memory due to the use of quadratic programming and complex algorithms [80].

3.12 Ordered Weighted Averaging Data Fusion

The ordered weighted averaging (OWA) was introduced by Ronald R.Yager (1988). This operator is commonly used in decision-making processes, which is still a powerful and simple method to finding aggregate output from data series. The OWA operator with n dimension is a mapping F: $R^n \rightarrow R$, and has an associated n vector, $w_i = (w_1, w_2, w_3, w_4, \dots, w_k)$.

Here n is the number of inputs that must be fused to achieve a final result. In the current study, n is equal to 14, because the database of seven wells times two classifiers (random forest and SVM) were utilized to discriminate fractured zones from non-fractured ones.

Weights are in the range of zero and one ($w_i \in [0, 1]$), and their summation have to be equal to 1 [81]:

$$\sum_{i=1}^{n} w_i = 1 \qquad (10\text{-}3)$$

In each depth, results achieved from different training wells and classifiers are defined as a_k. So, a_k could be zero (label which represents non-fractured zones) or one (label which represents fractured zones). Operator works as follows [81]:

$$F(\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{kn}) = \sum_{j=1}^{n} b_{kj} w_j = b_{k1} w_1 + b_{k2} w_2 + \dots + b_{kn} w_{jn}, k \qquad (11\text{-}3)$$

b_{kj} are in fact a_k which have ordered descending. Therefore, b_1 is the biggest label (usually 1) and b_{14} the least (usually 0). k represents the depths of the test well. Here, output of this cumulative operator is always in the range of zero and one, and could be said in the group of "OR" and "AND" [82, 83]:

$$Min_i\,[\alpha_i] \;\le\; F_w(\alpha_1, \alpha_2, \ldots, \alpha_n) \;\le\; Max_i\,[\alpha_i] \qquad (12\text{-}3)$$

Yager introduced an Orness evaluation that is related to vector addressed by w [82]:

$$orness\,(w) \;= \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i \qquad (13\text{-}3)$$

Therefore, for every "w", the Orness (w) is located at a unit distance. Suppose that d_k represents the real fracture label in each depth of the test well. So, errors should be minimized by optimizing the weights (w_i) [81]:

$$e \;=\; 1/2\left[\sum_{k=1}^{m}(b_{k1}w_1 \;+\; b_{k1}w_1 \;+\ldots+ b_{k1}w_1 \;- d_k)^2\right] \qquad (14\text{-}3)$$

The main problem is that the OWA operator must act everywhere in such a way that the error is minimized.

In other words, weights (w_i) must count in such a way that in the end, the error (e) goes towards zero. It will be obtained, while the solution is optimized. Yager has proposed two scenarios for optimization of the weights: optimistic, and pessimistic:

- **Optimistic Operator**

In optimistic ordered weighted averaging (OOWA), weights (w_i) are defined as a function of coefficient (α) [81]:

$$w_1 = \alpha;\ w_2 = \alpha(1-\alpha);\ w_3 = \alpha(1-\alpha)^2;\ \ldots;$$
$$w_{n-1} = \alpha(1-\alpha)^{n-2};\ w_n = (1-\alpha)^{n-1};\ 0 \le \alpha \le 1 \qquad (15\text{-}3)$$

Therefore, it is enough that α varies from zero to one and error (e) associated to different α be calculated. Minimum error, corresponds to optimum α, could be said optimum weights (w_i) in OOWA.

- **Pessimistic Operator**

In pessimistic ordered weighted averaging mechanism (POWA), weights (w_i) again are defined as a function of the α [81]:

$$w_1 = \alpha^{n-1};\ w_2 = (1-\alpha)\alpha^{n-2};\ w_3 = (1-\alpha)\ \alpha^{n-3};\ \ldots;$$
$$w_{n-1} = (1-\alpha)\alpha;\ w_n = (1-\alpha);\quad 0 \le \alpha \le 1 \qquad (16\text{-}3)$$

The procedure for optimization of weights (you say α), is similar to optimistic mechanism.

4

Implementation Procedure

4.1 Introduction

Nobody can answer the question about which machine learning algorithm is the best for a specific problem. It can just be found by applying different algorithms individually and comparing the results to figure out which one could work more accurately. DT, RF, SVM, and DL, are four classifiers that were used in the current study. The flowchart of procedures was followed up for FZ detection which is shown in Figure 4.1. Utilized methods are introduced briefly as follows:

4.2 Data Preprocessing

Data preprocessing is the first step in data mining. For this purpose, depth shifting, tolls pickup correction, tolls malfunction correction, cycle skipping data, and wash out error removing are done over all PLs data.

For two classifiers SVM and DL methods, data needs to be normalized. Relations 4-1 and 4-2 are the ways for normalization for SVM and DL respectively [84]:

$$\widehat{x_i} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \qquad \widehat{x_i} \rightarrow [0\,,1] \tag{4-1}$$

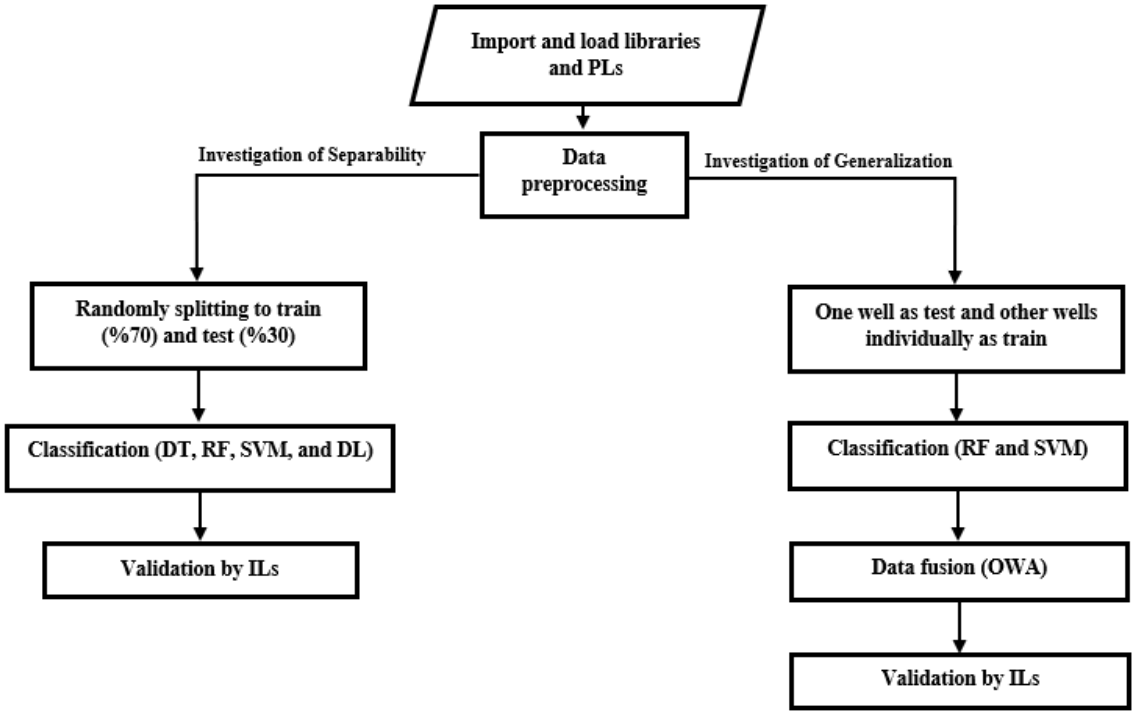


Figure 4.1 Steps for finding fractured zone by ML approaches

where $\widehat{x_i}$ is the normalization and xi is the primary amount of one of the logs at the same depth, and x_{min} and x_{max} are minimum and maximum range of data respectively [84]:

$$\widehat{\hat{x}_i} = 2\widehat{x_i} - 1, \qquad \widehat{\hat{x}_i} \to [-1, 1] \tag{4-2}$$

where $\widehat{\hat{x}_i}$, $\widehat{x_i}$ are normalized data.

4.3 Classifiers

4.3.1 Decision Tree and Random Forest

DT and RF follow the following algorithm in Scikit-learn in Python programing language:

a) Importing and loading required libraries such as numpy, pandas, xlsxwriter, matplotlib, seaborn, pylab, sklearn.model_selection, sklearn.tree, sklearn.metrics and klearn.ensemble.

b) Importing dataset (Petrophysics logs in the current study), by path direction and make the data as a data frame.

c) Dividing data into two types of variables. These two types are feature and target variables or dependent and independent variables.

d) Splitting data into train and test, by train_size=70% and test_size=30%.

e) Creating a DT model and an RF model by using Scikit-learn and fitting the model, the number of estimators can be varied, and the best one can be defined by several tries and test and find the best result that is precise..

f) Evaluating the model by how accurately the classifier or model can predict the type of samples. This is done by comparing actual test set values and predicted values. Commands classification_report and confusion_matrix show the correct classification rate (CCR), precision, recall, f1-score, and support values.

g) Improving accuracy by tuning the parameters such as the criterion in the Algorithm. In this research, Gini Index is defined as a criterion. Optimizing performance by a) changing the attribute selection measure, b) changing the split strategy, and c) changing the maximum depth of a tree.

In appendix A and B, codes that are written for DT and RF are shown respectively.

- ***Visualization of DT:***

In this study for splitting Decision Tree nodes, Gini Index has been used. The Gini Index varies between 0 and 1, where 0 represents purity of the classification and 1 denotes random distribution of elements among various classes. This is a criterion for categorical and splits variables to the binary classes.

Gini Index works based on the concept of entropy which is used to determine which attribute gives us the maximum information about a class. Entropy is the degree of impurity or uncertainty. This index tries to decrease the level of this impurity from the root nodes to the leaf nodes of the decision tree.

In each node, the Gini Gain is calculated for each feature in the dataset. The comment sklearn.tree. DecisionTreeClassifier, will select the largest Gini Gain as the Root Node. If Gini reaches 0, then the branch becomes the leaf node. By Gini, more than 0 splitting is continuing.

These nodes are grown recursively until all of them are classified [85].

In Fig 4.2 the visualize of the Decision Tree for well 1 is shown. For more clarity, Fig 4.3 is shown.

As it shows, the Decision Tree for 70% of the total data set is created. It is a training part for classification. Each node contains a feature or attributes that are present in one of the petrophysical logs data (X[index]), which the index is PL data. Each node is separated in half True (Is the feature less or equal than the range that the Gini Index is higher?) or false (Is the feature less or equal than the range that the Gini Index is lower?) and how much of the sample data in this node belonged to this condition.

In appendix C, the code that is written for DT visualization is shown.

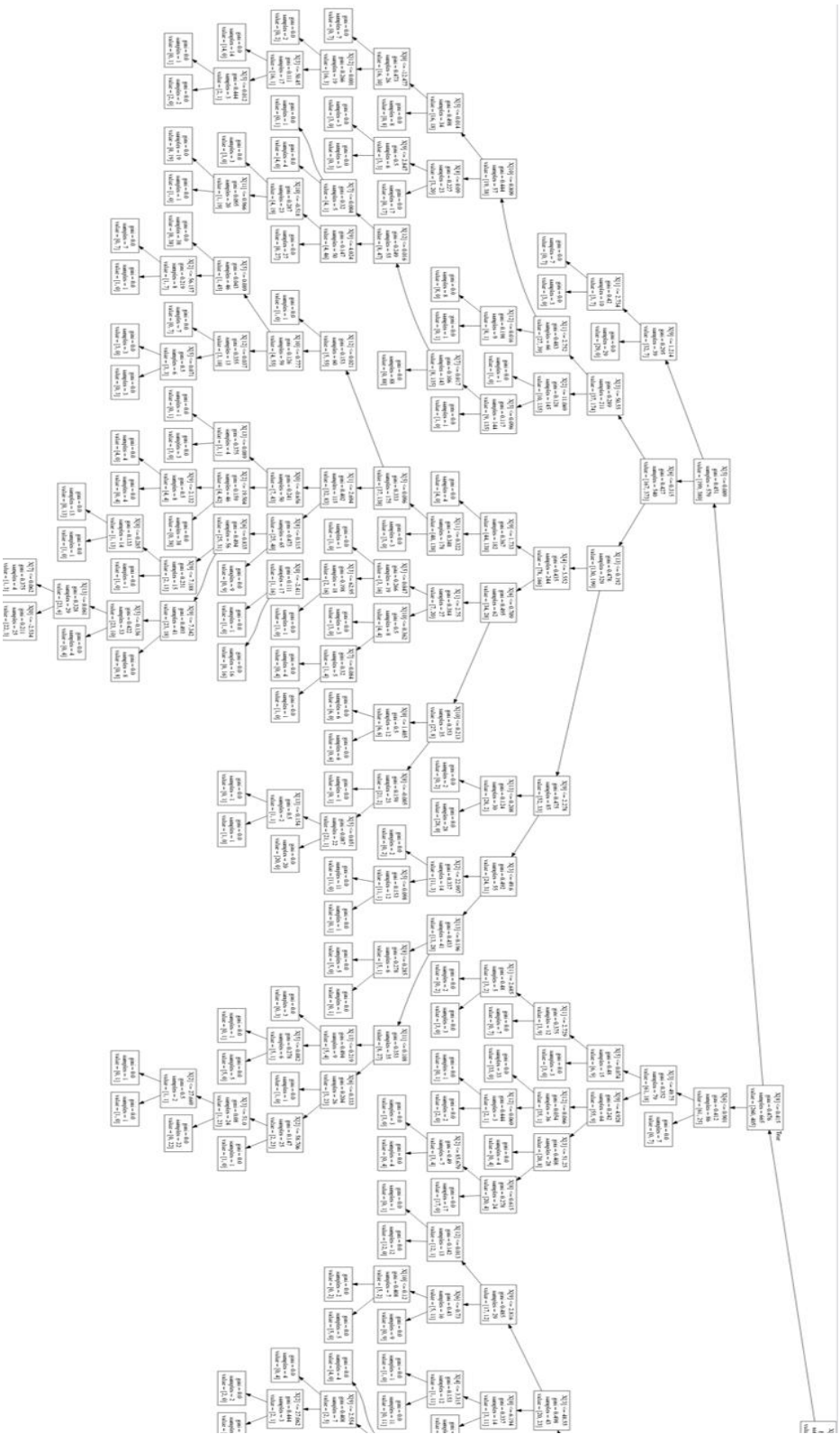


Figure 4.2. Output of Decision Tree for well 1.

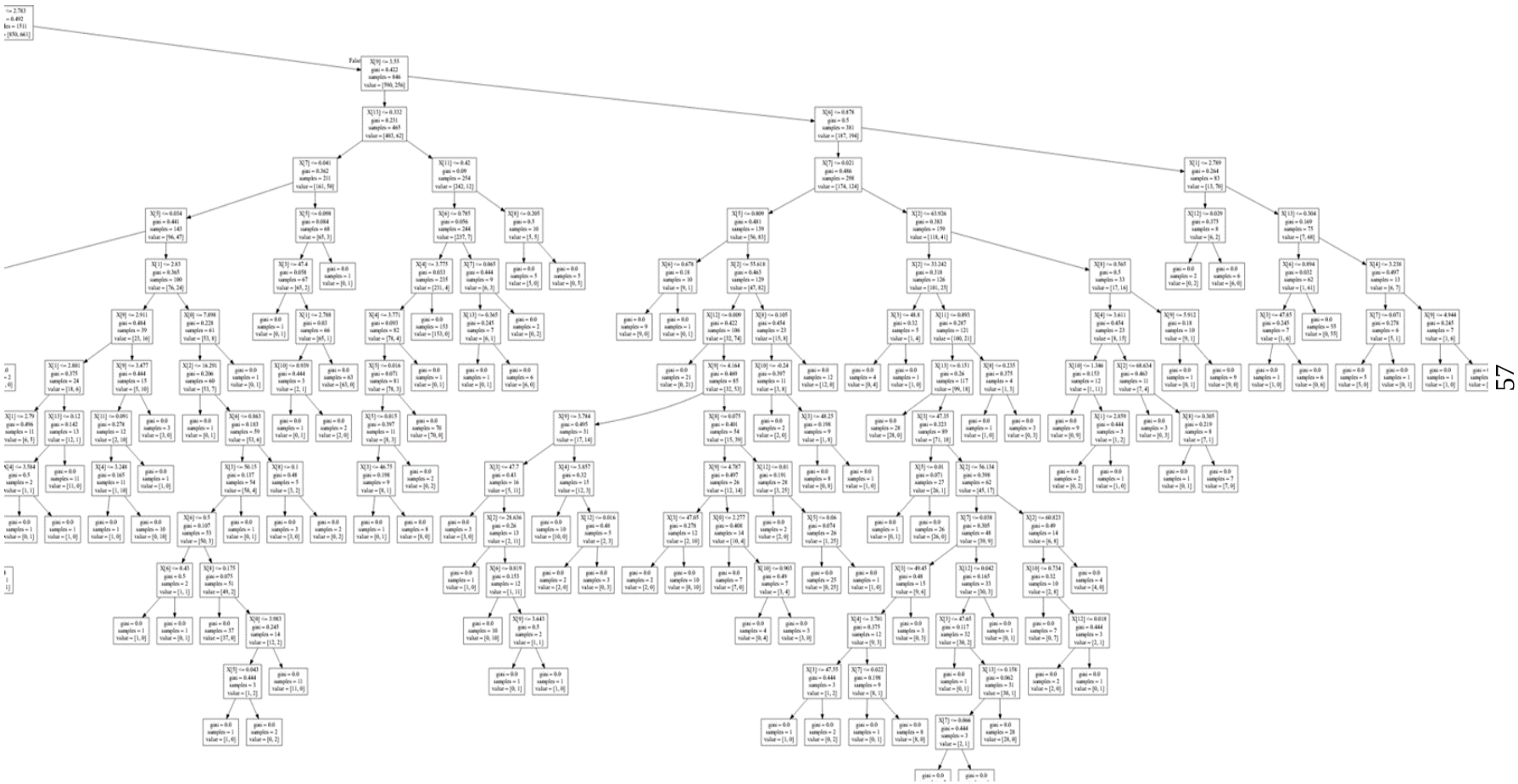
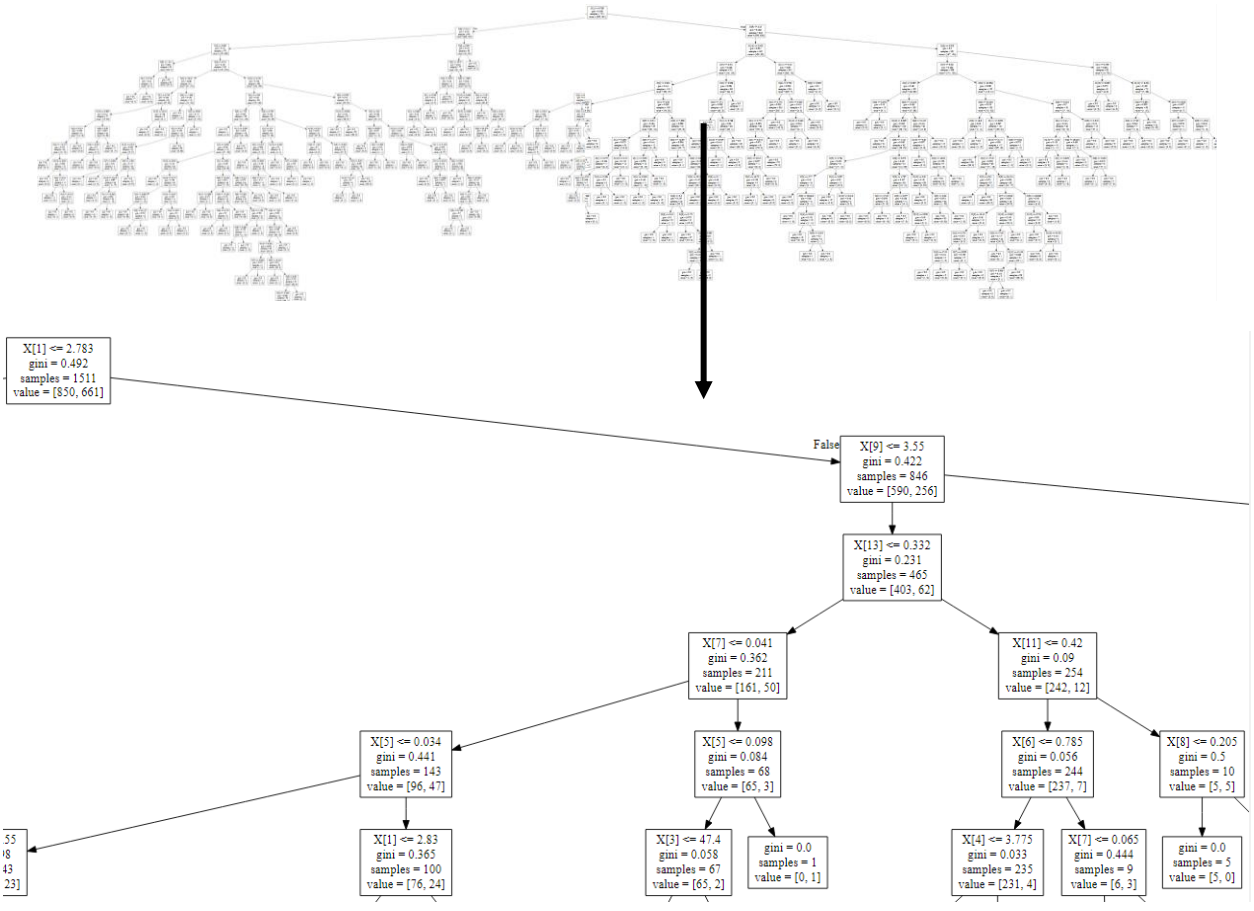


Figure 4.2. Output of Decision Tree for well 1.



X[0]	X[1]	X[2]	X[3]	X[4]	X[5]	X[6]	X[7]	X[8]	X[9]	X[10]	X[11]	X[12]	X[13]
CGR	RHOB	SGR	DT	PEF	NPHI	DOLOM	SHALE	POTA	URAN	THOR	LIME	PORE	SW

Figure 4.3. Output of training Decision Tree by using 70 % of the data of well 1. Index of PLs and their range are reported. Training continues by the time all fractured zones were discriminated from non-fractured ones. In this case by running a trained classifier over test data, accuracy (CCR) was about 80 percent.

4.3.2 Support Vector Machine

The SVM algorithm is very similar to the DT and RF one. Differences will be such as:

a) Importing the SVC from sklearn.svm. in step one and fitting the SVM model on data in step five.

b) For the last step, SVM classifier and parameter is optimized by cross-validation grid-search to find the best model. In this study, to do optimization, SKlearn's GridSearchCV is used.

Also, the Radial Basis Function (RBF) as a kernel hyperparameters is used. RBF is suitable for this research since PLs data set are linearly inseparable or non-linear.

On the other hand, choosing the best fit kernel is a big issue in which the wrong choice of kernel can lead to higher error. Although SVMs are a popular method, they might need a big runtime, especially in big data problems. Also, they need extensive memory due to the use of quadratic programming and complex algorithms.

In appendix D, the code that is written for SVM is shown.

4.3.3 Deep Learning

In this study, the Multilayer Perceptron is implemented in the Colab. Google Colab is a free cloud service and now it supports free GPU and has popular libraries for developing DL applications such as Kersa, TensorFlow, PyTorch, and OpenCV. The steps of implementation are like DT, RF, and SVM with some differences such as follows:

a) Import libraries (google.colab, io, sklearn.preprocessing, tensorflow, tensorflow.keras) and datasets

b) Initialize model parameters (number of class (here binary classification), number of features (here are 10 features), number of hidden layers (we tried several nodes in 3 hidden layers), and bias vector)

c) Determine the activation function (here tf.nn.relu is used)

d) Determine the model (here model_dir = 'models/iris')

e) Implement the Loss function to examine the source code to deepen the information of implementation details. Backpropagation performs iterative backward passes which

attempt to minimize the “loss”. It optimizes the weights and minimizes the difference between the real results and predicted results. Here loss_reduction=losses_utils and optimizer= lambda: tf.keras.optimizers.Adam.

f) Train the model on data. It depends on the number of steps necessary to train the model. If None, train forever or train until input_fn generates the tf.errors.OutOfRange error or StopIteration exception. In this study, we didn’t want to have incremental behavior so we set the steps to equal 200.

g) Evaluated values of predictions tensors. Here is eval_results3 = model.evaluate (input_fn=lambda:eval_input_fn(X_test, y_test), steps=1).

In appendix E, the code that is written for DL is shown.

After applying classifiers, Ambiguity in decision making leads to data fusion. The data fusion method takes advantage of ordered weighted averaging aggregation operator and fuzzy defined interval to fuse the results achieved from different classifiers and all wells to obtain more accurate and reliable results. The results of the experiments revealed that applying data fusion to the outputs obtained from different classifiers produces more accurate discrimination compared to the situation that only outputs of one classifier are used.

4.4 Ordered Weighted Averaging

Table 4.1 shows an example of how the OWA method is used for data fusion. Fractures in every depth are shown by number 1. RF1, FR2, SVM1, and SVM2 are classifiers that are used on this part of the well test data while other wells’ data are a training one. For fusing the results of these four classifiers, two methods are introduced. The first one is Majority Voting (MV), that is a decision rule that selects alternatives which have a majority, that is, more than half the votes. And

the second one is OWA. As Table 4.1 shows, OWA makes a better fusion than MV. For example, while MV leads to an uncertainty zone, OWA can fuse results correctly as a fractured zone (True Positive). Also, in the part that MV shows False Negative, OWA can make a better fusion. Therefore, this study using OWA on classification results achieved reliable and robust results.

Table 4.1 An example of beneficiary of using OWA data fusion for integration of results to achieve more reliable/accurate/general discrimination

DEPTH	Fracture	RF1	RF2	SVM1	SVM2	MV	OWA	Interpretation Majority Voting	Interpretation OWA
2369.972	0	0	0	0	0	0	0	Non-Fractured Zone **(True Negative)**	Non-Fractured Zone **(True Negative)**
2370.125	0	0	0	0	0	0	0		
2370.277	0	0	0	0	0	0	0		
2370.43	0	0	0	0	0	0	0		
2370.582	1	0	0	1	1	0.5	0.8	Uncertainty Zone **(Uncertainty)**	Fractured Zone **(True Positive)**
2370.734	1	0	0	1	1	0.5	0.8		
2370.887	1	0	0	1	1	0.5	0.8		
2371.039	1	1	1	1	1	1	1	Fractured Zone **(True Positive)**	Fractured Zone **(True Positive)**
2371.192	1	1	1	1	1	1	1		
2371.344	1	1	1	1	1	1	1		
2371.496	1	1	1	1	1	1	1		
2371.649	1	1	1	1	1	1	1		
2371.801	1	1	0	0	0	0	0.56	Non-Fractured Zone **(False Negative)**	Fractured Zone **(True Positive)**
2371.954	1	1	0	0	0	0	0.56		
2372.106	1	1	0	0	0	0	0.56		
2372.258	1	1	0	0	0	0	0.56		
2372.411	1	1	0	0	0	0	0.56		

Fig 4.4 is a schematic way of validation of classifications. In this figure, the image log works as a validation criterion in which fractured and non-fractured zones are determined by professional interpretation image processing. In the next of image log, a schematic classified zone is shown. By cross-validation between real and classified fractured zones, True Positive, False Negative, False Positive, and True Negative will be achieved. On the other hand, this leads to the configuration of the confusion matrix which is the base of all validation.

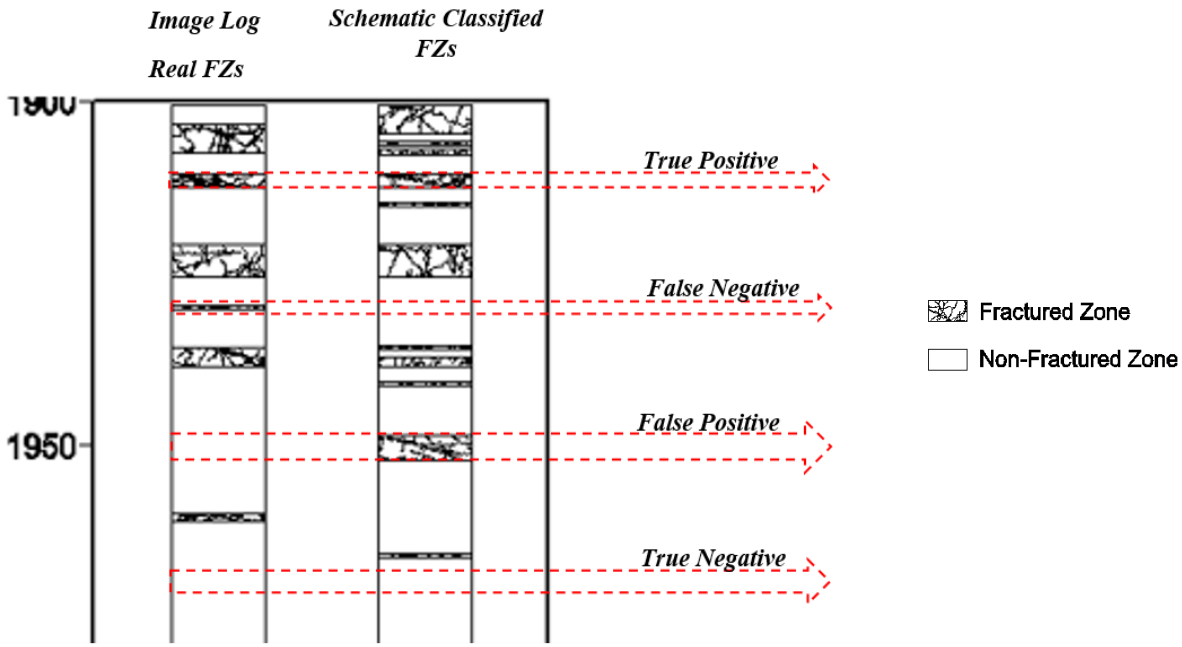


Figure 4.4 A schematic validation procedure

5

Experimental Results: Fractured Zone Detection from Petrophysical Logs

5.1 Introduction

As aforementioned, the databases of eight wells were used to investigate how fractured zones could be discriminated from non-fractured zones by using petrophysical logs. Petrophysical logs as well as image logs were available for studied wells. Image logs were professionally interpreted, and fractures and their properties, e.g. depth, opening, aperture, dip and dip direction were reported. By using the depth of the fractures, wells were labeled with zero for non-fractured zones, and one for fractured zones. Therefore, a valuable database was achieved containing petrophysical logs and fracture labels.

In this chapter, the correlation between all logs, depth and fracture is reported. This helped in log selection. Histograms of selected logs as well as 2D and 3D graphs for fractured versus non-fractured zones are plotted to investigate which logs are useful for fractured zone detection.

Afterward, the results of applying a two-step procedure, containing classification and data fusion for the discrimination of fractured/non-fractured zones are also represented. Those procedures were applied over imbalanced and balanced versions of the petrophysical logs.

5.2 Statistical Studies

Table 5.1 shows cross correlation between all logs plus fractured zone and depth in well number 1. As it shows, correlations between fractures and petrophysics logs are not impressive, while its maximum amount is about 0.25. Therefore, it seems discrimination between fractured/non-fractured zones is not straight-forward.

Correlation between fractured zones and Water Saturation (SW) is negative, which means fractures are filled by oil instead of water. Positive correlation between porosity (POR) and fractures is an indicator that fractures are mainly open.

Limestone (LIME) and dolomite both are brittle rocks, therefore their correlation with fracture might be positive. This has happened for limestone, while for dolomite, correlation is about zero, because it has not been the main frame of rock in the studied reservoir.

Thorium and potassium are primarily radioactive materials, formed during sedimentation by shale, while uranium (URAN) is a secondary accumulated material that migrates by fluid to the reservoir. Fractures are a suitable path for fluid flow and deposition of the uranium. Therefore, a positive correlation between uranium and fractured zones could also be reasonable. The spectral gamma-ray (SGR) is a summation of potassium, thorium and uranium. In carbonate reservoirs, limestone is the dominant rock, which means the lack of primarily radioactive materials. Therefore, SGR would be mainly affected by uranium and their high correlation (0.94) confirms this hypothesis. Its correlation of fractures is also similar to uranium.

Table 5.1: Cross correlation between all petrophysical logs, fracture label and depth in well 1

	DEPTH	Fracture	CALI	CGR	RHOB	SGR	DT	PEF	NPHI	DOLOMITE	SHALE	POTA	URAN	THOR	LIME	PORE	SW
DEPTH	1	-0.0044	-0.5098	-0.4004	-0.0531	-0.3433	0.0299	0.3133	-0.0602	-0.2534	-0.2460	-0.3887	-0.2089	-0.2863	0.3424	0.0637	-0.1928
Fracture	-0.0044	1	0.1088	-0.2072	-0.2216	0.1225	0.1420	0.0259	0.0088	-0.02273	-0.2103	-0.2251	0.2033	-0.1242	0.2267	0.156334	-0.2479
CALI	-0.5098	0.1088	1	0.1628	-0.2035	0.3075	-0.0012	-0.2683	0.0491	0.204364	0.0119	0.1352	0.2419	0.1170	-0.1469	0.018464	-0.1007
CGR	-0.4004	-0.2072	0.1628	1	-0.0478	0.3547	0.0186	-0.2668	0.1774	0.178001	0.8336	0.9246	0.0129	0.8164	-0.2480	-0.1664	0.2099
RHOB	-0.0531	-0.2216	-0.2035	-0.0478	1	-0.1055	-0.6114	0.0628	-0.5961	0.060154	-0.1436	-0.0209	-0.0879	-0.0556	-0.3754	-0.58777	0.3670
SGR	-0.3433	0.1225	0.3075	0.3547	-0.1055	1	-0.1012	-0.4349	0.1802	0.444305	0.2707	0.3332	0.9390	0.2700	-0.3076	0.023002	-0.1178
DT	0.0299	0.1420	-0.0012	0.0186	-0.6114	-0.1012	1	-0.0204	0.6513	-0.11206	0.1853	0.0348	-0.1165	-0.0156	0.2188	0.742213	-0.3243
PEF	0.3133	0.0259	-0.2683	-0.2668	0.0628	-0.4349	-0.0204	1	-0.5054	-0.93244	-0.1475	-0.2733	-0.3626	-0.1690	0.7824	-0.4338	0.4304
NPHI	-0.0602	0.0088	0.0491	0.1774	-0.5961	0.1802	0.6513	-0.5054	1	0.4319	0.3581	0.1930	0.1262	0.0962	-0.2563	0.863104	-0.4976
DOLOMITE	-0.2534	-0.0227	0.2043	0.1780	0.0601	0.4443	-0.1120	-0.9324	0.4319	1	0.0553	0.1839	0.4073	0.1106	-0.8371	0.390404	-0.4510
SHALE	-0.2460	-0.2103	0.0119	0.8336	-0.1436	0.2707	0.1853	-0.1475	0.3581	0.055373	1	0.8515	-0.0117	0.5620	-0.1183	-0.02278	0.0912
POTA	-0.3887	-0.2251	0.1352	0.9246	-0.0209	0.3332	0.0348	-0.2733	0.1930	0.183956	0.8515	1	0.0185	0.5358	-0.2717	-0.15667	0.1986
URAN	-0.2089	0.2033	0.2419	0.0129	-0.0879	0.9390	-0.1165	-0.3626	0.1262	0.407353	-0.0117	0.0185	1	-0.0105	-0.2372	0.083704	-0.2004
THOR	-0.2863	-0.1242	0.1170	0.8164	-0.0556	0.2700	-0.0156	-0.1690	0.0962	0.110614	0.5620	0.5358	-0.0105	1	-0.1382	-0.13655	0.1734
LIME	0.3424	0.2267	-0.1469	-0.2480	-0.3754	-0.3076	0.2188	0.782	-0.2563	-0.83713	-0.1183	-0.2717	-0.2372	-0.138	1	-0.15400	0.1479
PORE	0.0637	0.156334	0.018464	-0.1664	-0.587778	0.023018	0.742213	-0.433	0.8631	0.390416	-0.0228	-0.1567	0.083778	-0.1365	-0.1540	1	-0.632
SW	-0.1928	-0.2479	-0.1007	0.2099	0.3670	-0.1178	-0.3243	0.4304	-0.4976	-0.45108	0.0912	0.1986	-0.2004	0.1734	0.1479	-0.6325	1

Cumulative gamma ray (CGR), is the summation of potassium and thorium, and an indicator of shale. Hence, shale, CGR, potassium and thorium, all have to have a similar correlation with fractures. It should be mentioned that shale is a ductile and in fact plastic rock. Therefore, unlike

the limestone, it is anticipated that its correlation to fracture be negative. Ironically, correlation of shale, CGR, potassium and thorium to fractures are negative, which is reasonable.

NPHI is an indicator for hydrogen, which is a main part of water and oil as well. Based on literature, and the image log's interpretation, a majority of the fractures in the studied reservoir are open, while the rest are closed. Therefore, a combination of occurrence of open and closed fractures is the reason that NPHI's correlation with fracture is low.

Photoelectric factor (PEF), is an indicator of specific weight. In oil wells, PEF usually increases because of deposition of mud cake, which contains barite with a specific weight higher than 4 gr/m3 in fractures or porous media. Therefore, its correlation with fractures is also positive. Density log (RHOB) is mainly affected by rock and fluid. By increasing the fluid saturation, it might be decreased. Hence, in fractured zones, density should decrease, which negatively affects the correlation between RHOB and fractures confirms it.

Sonic (DT) is addressed as a suitable indicator for fractures. The reason is that in fractured zones, the sonic wave arrival time will increase. Also, its correlation with fractures is positive. In the studied reservoir, two leg caliper logs were run, which could not be an effective indicator of fractured zones. The reason is that it could be decreased because of mud cake, or increased because of washout around open fractures.

Tables 5.2 and 5.3 show correlation coefficient between logs, fracture and depth in wells 3 and 5 respectively. They, overall, confirm the observations in well 1. However, minor differences are a sign of complexity of discrimination of fractured zones from non-fractured ones.

In appendix F, the code that is written for correlation coefficient between logs, fracture and depth is shown.

5.3 Histograms

Comparison between the histogram of petrophysical logs in fractured/non-fractured zones will help to figure out which logs might be useful to discriminate those zones from each other. These histograms and corresponding probability density functions (PDFs) for well 1 are displayed in Figure 5.1. As it can be seen in Figure 5.1, almost all the logs of the histograms of two classes are matched together, which clearly shows why the industry doesn't utilize petrophysical logs for fractured zone detection. Also, it shows that discrimination between fractured/non-fractured zones in 1D feature space (by using just one log) is impossible. In appendix G, similar histograms for petrophysical logs of wells 2 and 5 are presented, which again show the same behavior as well 1. However, it might be possible that by increasing feature space and using more logs, classification of fractured/non-fractured zones is possible. To visually investigate this possibility, 2D and 3D plots of fractured/non-fractured zones should be plotted.

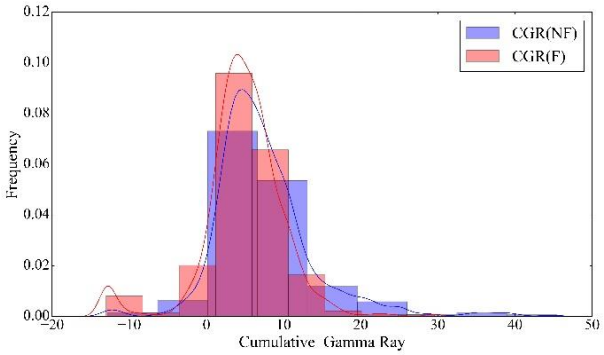
Code that is written for Histogram and the corresponding PDF for different logs in fractured zones and non-fractured ones is shown in appendix H.

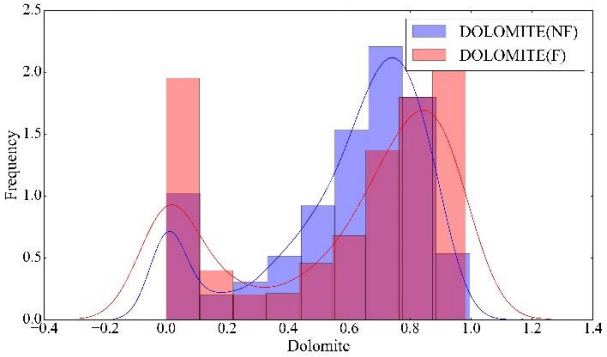
Table 5.2 Cross correlation between all petrophysical logs, fracture label and depth in well 3

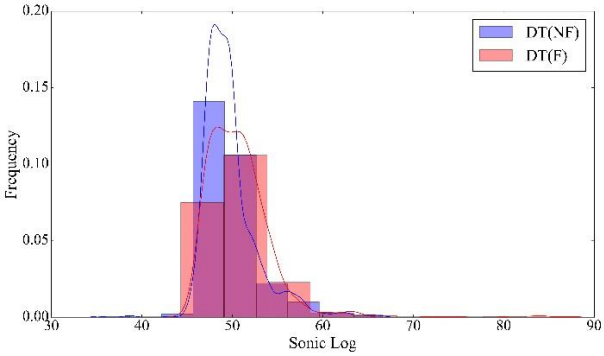
	DEPTH	Fracture	CALI	CGR	RHOB	SGR	DT	PEF	NPHI	POTA	URAN	THOR
DEPTH	1	-0.07336	-0.57776	-0.09963	-0.05912	-0.32121	0.227034	-0.0844	0.161339	-0.11378	-0.32386	-0.03263
Fracture	-0.07336	1	-0.07464	0.013317	-0.16876	0.149943	0.198929	-0.28905	0.154407	0.022499	0.161401	-0.00121
CALI	-0.57776	-0.07464	1	0.170612	0.041254	0.123247	-0.1629	0.068334	-0.18031	0.163828	0.081759	0.091154
CGR	-0.09963	0.013317	0.170612	1	-0.01025	0.436472	-0.04695	-0.10667	-0.07539	0.859329	0.142732	0.766563
RHOB	-0.05912	-0.16876	0.041254	-0.01025	1	-0.03041	-0.78033	0.6722	-0.9072	-0.03469	-0.02825	0.013941
SGR	-0.32121	0.149943	0.123247	0.436472	-0.03041	1	-0.04126	-0.22342	-0.02338	0.398561	0.948783	0.367711
DT	0.227034	0.198929	-0.1629	-0.04695	-0.78033	-0.04126	1	-0.46904	0.824402	-0.03039	-0.03018	-0.04583
PEF	-0.0844	-0.28905	0.068334	-0.10667	0.6722	-0.22342	-0.46904	1	-0.63973	-0.12778	-0.20908	-0.04885
NPHI	0.161339	0.154407	-0.18031	-0.07539	-0.9072	-0.02338	0.824402	-0.63973	1	-0.04937	-0.00321	-0.06076
POTA	-0.11378	0.022499	0.163828	0.859329	-0.03469	0.398561	-0.03039	-0.12778	-0.04937	1	0.128948	0.435153
URAN	-0.32386	0.161401	0.081759	0.142732	-0.02825	0.948783	-0.03018	-0.20908	-0.00321	0.128948	1	0.122347
THOR	-0.03263	-0.00121	0.091154	0.766563	0.013941	0.367711	-0.04583	-0.04885	-0.06076	0.435153	0.122347	1

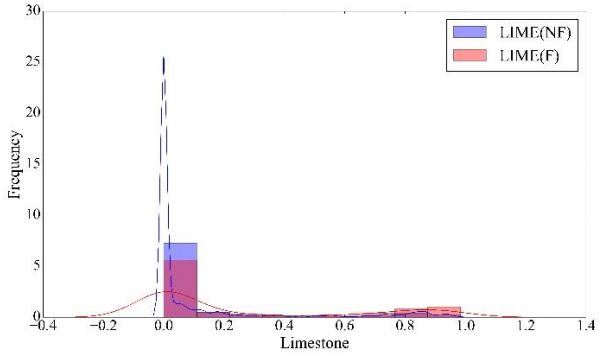
Table 5.3 Cross correlation between all petrophysical logs, fracture label and depth in well 5

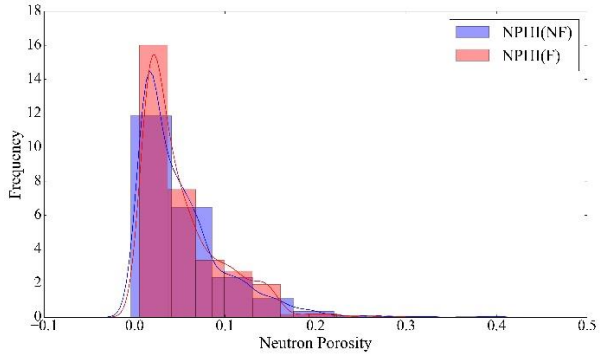
	DEPTH	Fracture	CALI	CGR	RHOB	SGR	DT	PEF	NPHI	DOLOMITE	SHALE	POTA	THOR	PORE	PORT
DEPTH	1	0.09450	0.02718	0.11906	0.71467	-0.43681	-0.60325	0.57777	-0.62173	-0.44189	0.17223	0.00678	0.00081	-0.24934	-0.72354
Fracture	0.09450	1	-0.02503	0.05329	-0.15099	0.16103	0.15057	-0.03219	0.14026	0.14232	0.02419	-0.04291	0.05300	-0.11722	0.07144
CALI	0.02718	-0.02503	1	0.01329	0.11686	-0.04654	-0.03725	0.35799	-0.09773	-0.09184	0.07723	-0.00261	-0.11511	-0.0954	-0.11862
CGR	0.11906	0.05329	0.01329	1	-0.00874	0.31428	0.22328	0.00841	0.19750	0.149966	0.69630	0.01674	0.83116	0.16054	-0.11249
RHOB	0.71467	-0.15099	0.11686	-0.00874	1	-0.62462	-0.91009	0.63065	-0.91235	-0.68264	0.04746	0.04794	-0.10234	-0.14695	-0.85807
SGR	-0.43681	0.16103	-0.04654	0.31428	-0.62462	1	0.72321	-0.27101	0.73690	0.677047	0.14975	0.16413	0.48010	0.17527	0.54574
DT	-0.60325	0.15057	-0.03725	0.22328	-0.91009	0.72321	1	-0.54896	0.96253	0.705118	0.18042	-0.0614	0.30233	0.13331	0.80990
PEF	0.57777	-0.03219	0.35799	0.00841	0.63065	-0.27101	-0.54896	1	-0.54579	-0.32854	0.00378	0.00175	-0.10625	-0.14119	-0.56335
NPHI	-0.62173	0.14026	-0.09773	0.19750	-0.91235	0.73690	0.96253	-0.54579	1	0.738413	0.12858	-0.06442	0.30043	0.27865	0.83794
DOLOMITE	-0.44189	0.14232	-0.09184	0.14996	-0.68264	0.67704	0.70511	-0.32854	0.73841	1	0.01258	-0.01774	0.21917	0.088604	0.561304
SHALE	0.17223	0.02419	0.07723	0.69630	0.04746	0.14975	0.18042	0.00378	0.12858	0.012586	1	-0.02833	0.59494	0.17421	-0.24177
POTA	0.00678	-0.04291	-0.00261	0.01674	0.04794	0.16413	-0.0614	0.00175	-0.06442	-0.01774	-0.02833	1	0.01469	-0.0315	-0.0543
THOR	0.00081	0.05300	-0.11511	0.83116	-0.10234	0.48010	0.30233	-0.10625	0.30043	0.21917	0.59494	0.01469	1	0.25563	0.03879
PORE	-0.24934	-0.11722	-0.0954	0.16054	-0.14695	0.17527	0.13331	-0.14119	0.27865	0.088604	0.17421	-0.0315	0.25563	1	0.32088
PORT	-0.72354	0.07144	-0.11862	-0.11249	-0.85807	0.54574	0.80990	-0.56335	0.83794	0.561304	-0.24177	-0.0543	0.03879	0.32088	1

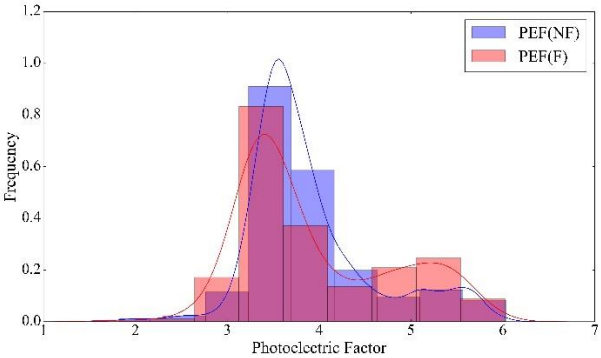

CGR(NF)
CGR(F)
Frequency
Cumulative Gamma Ray

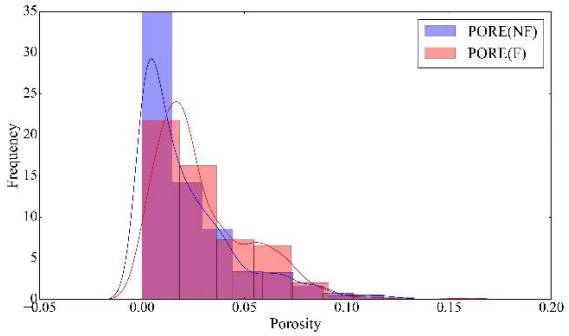

DOLOMITE(NF)
DOLOMITE(F)
Frequency
Dolomite

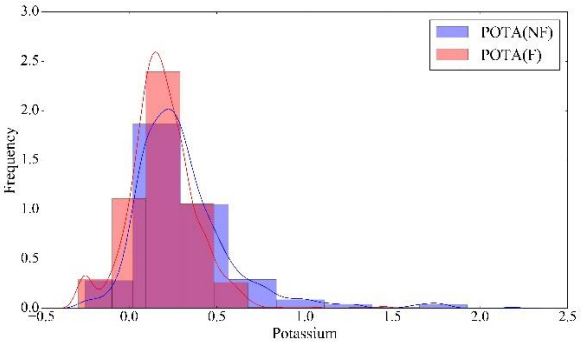

DT(NF)
DT(F)
Frequency
Sonic Log


LIME(NF)
LIME(F)
Frequency
Limestone


NPHI(NF)
NPHI(F)
Frequency
Neutron Porosity


PEF(NF)
PEF(F)
Frequency
Photoelectric Factor


PORE(NF)
PORE(F)
Frequency
Porosity


POTA(NF)
POTA(F)
Frequency
Potassium

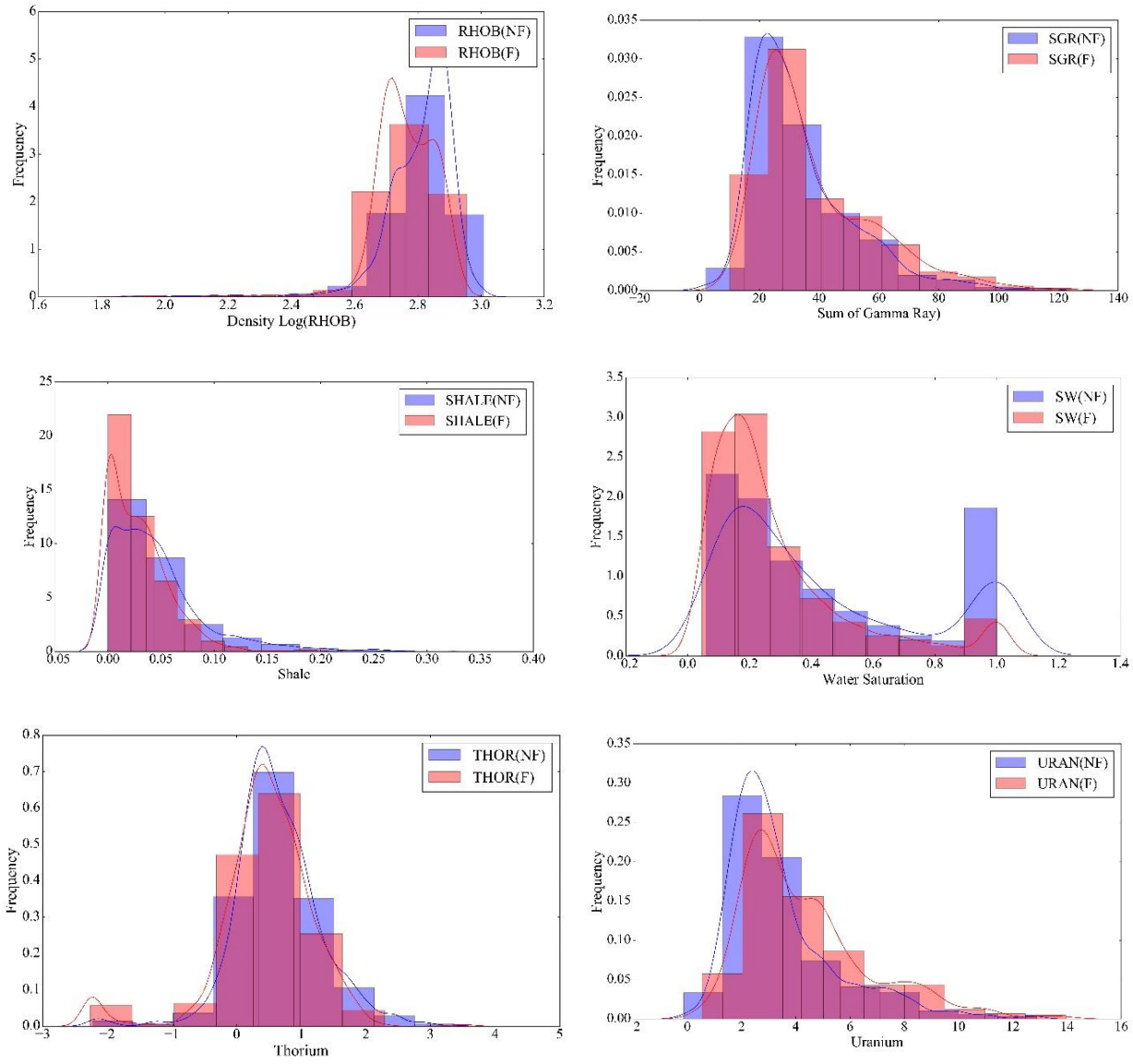


Figure 5.1 Histogram and corresponding PDF for different logs in fractured zones and non-fractured ones in well 1.

5.4 Petrophysical Log Selection

Availability of logs, their correlation coefficient with fractured zones, as well as studies were done over histograms and helped to log selections for fractured zone detection. Selected logs, and the reason they have been selected are described as follows:

Correlation coefficient of caliper with fractures is low; however, it was selected, because it is an individual log with low correlation to other logs. Caliper is also available in all studied wells. It might be useful for fractured zone detection.

Correlation coefficients between CGR, potassium, thorium and shale are high, therefore, one of them could represent the others. CGR was selected for the study just because it is available in all of the studied wells. SGR is also preferred over uranium, because it is available in more wells.

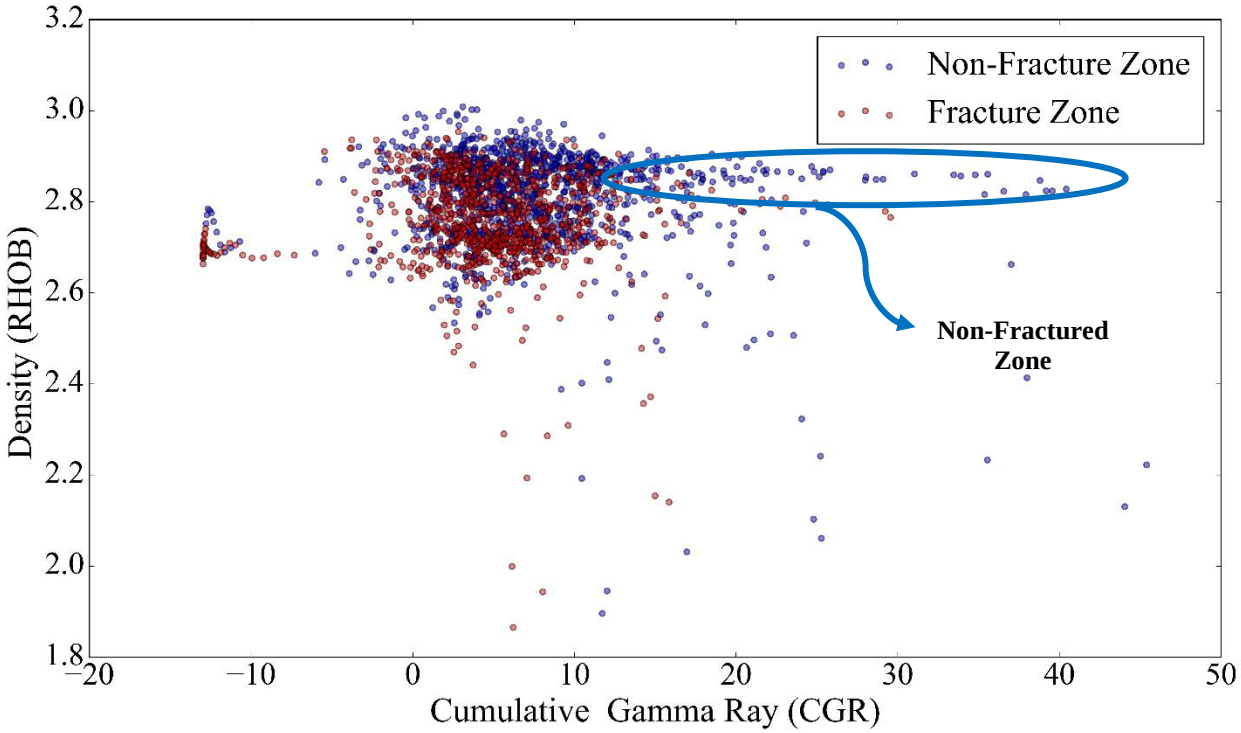
Effective porosity could be efficient for the discrimination of open fractures from closed ones, which of course is not the aim of this study. However, both effective and total porosity were selected.

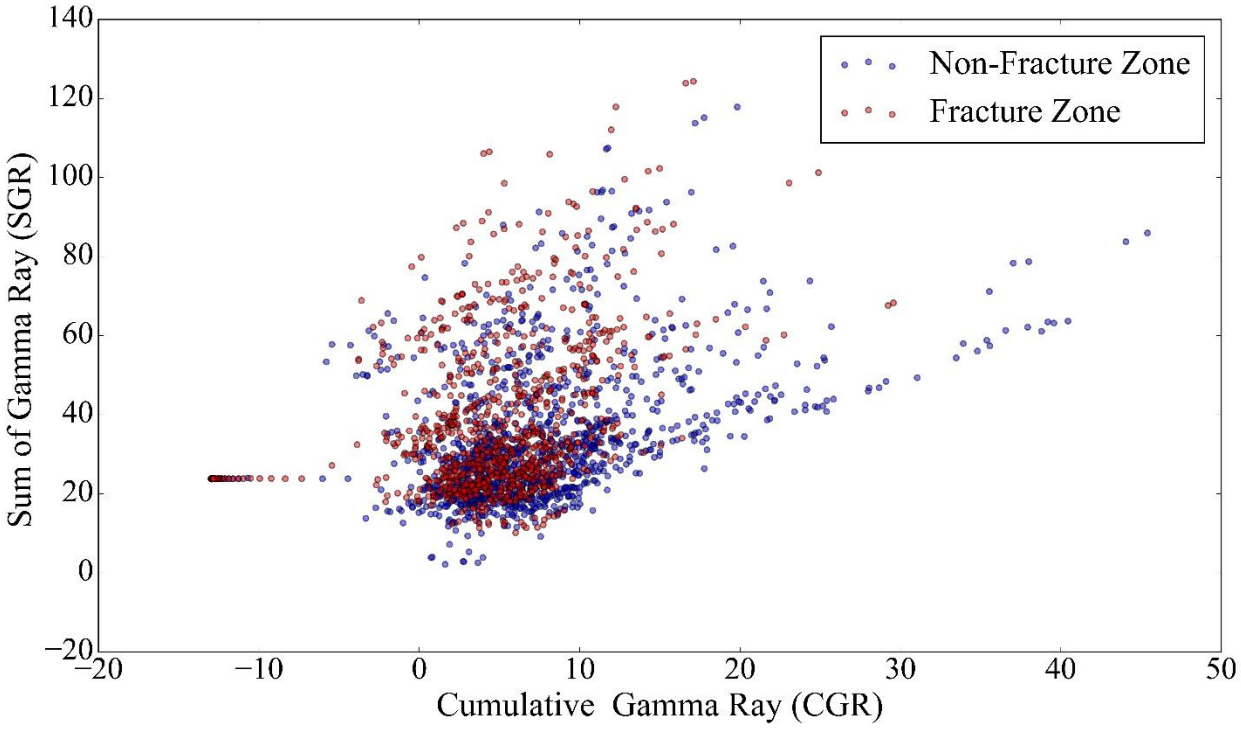
Dolomite and limestone might be useful, but weren't selected, because they are not available in a majority of wells. SW, RHOB, DT, PEF and NPHI were also selected, because of their correlation of fractures and availability as well.

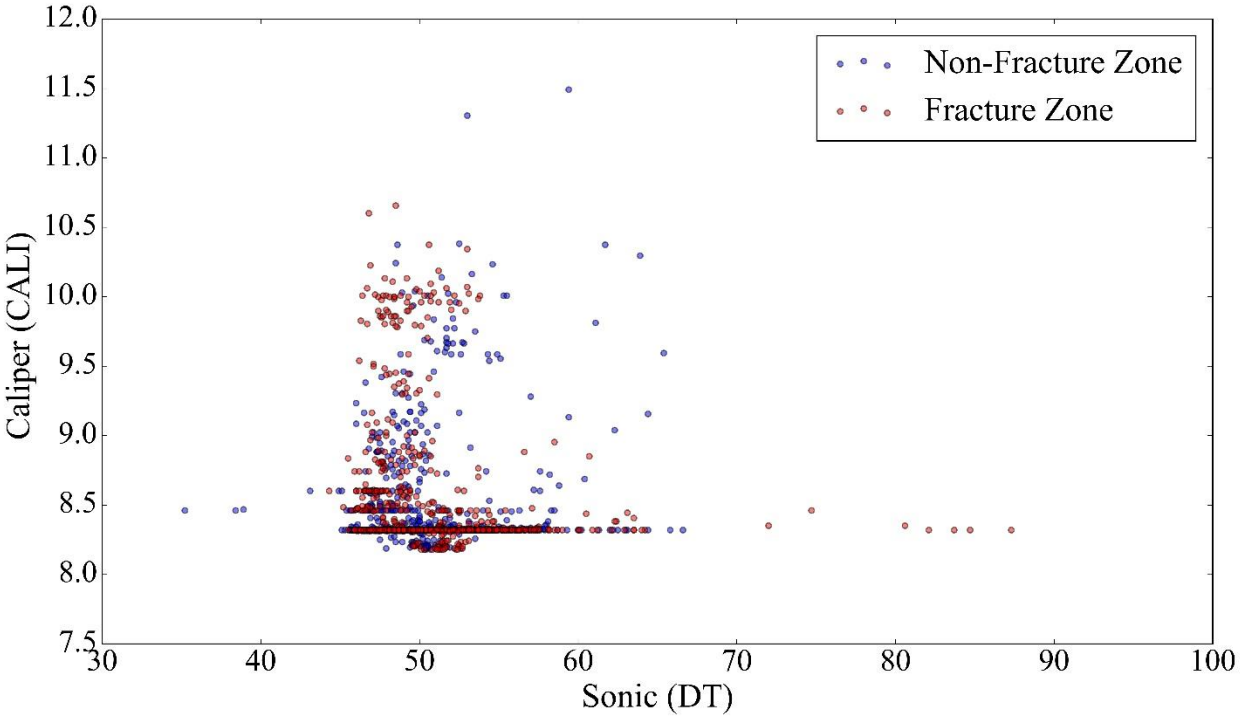
Overall, caliper, CGR, SGR, RHOB, DT, PEF, NPHI, SW, effective and total porosity were selected for fractured zone detection. Therefore, classifiers will define and discriminate fractured/non-fractured zones in 10D feature space.

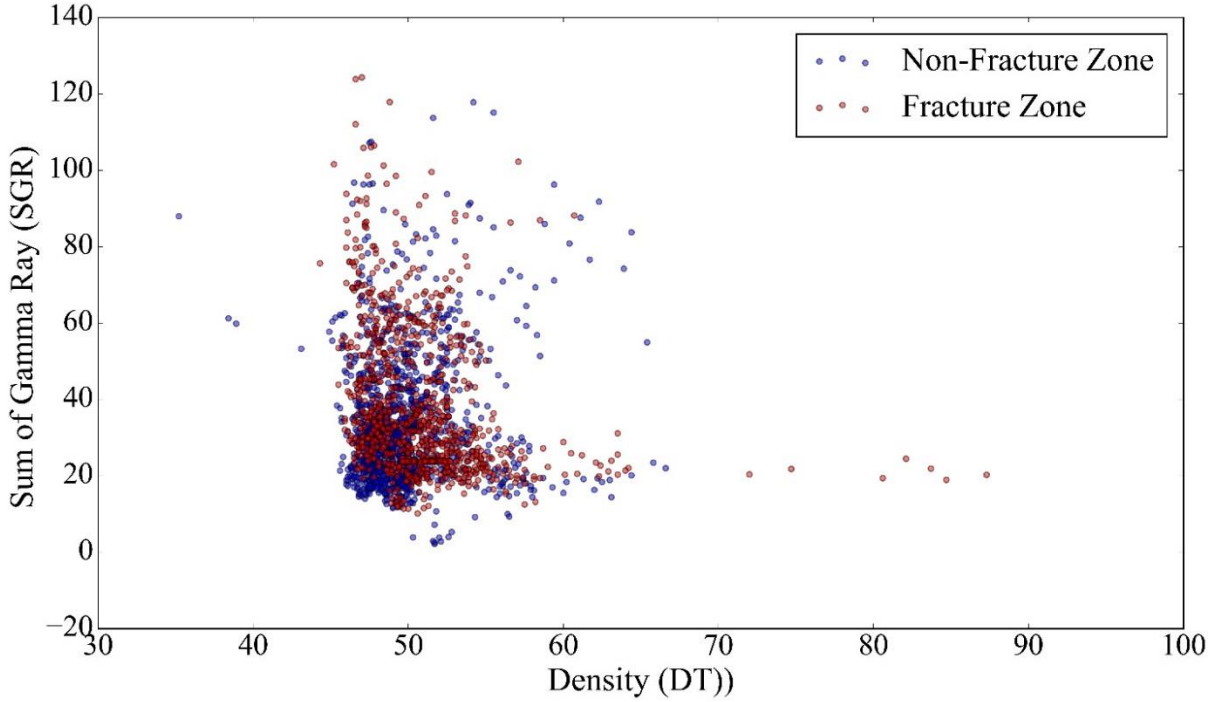
5.5 2D and 3D Cross Plot of Petrophysical Log

2D and 3D cross plots of petrophysical logs could visually show the importance of feature space over discrimination of fractured zones from non-fractured ones. In Figure 5.2, 2D cross plots in well 1 for selected logs are displayed. Comparison between Figures 5.1 and 5.2 clearly shows better discrimination in 2D feature space. For example, high CGR and low RHOB contains mainly non-fractured zones. Similar interpretation could be presented for other cross plots. In appendix I, similar cross plots for wells 2 and 5 are displayed and the corresponding code is in appendix J.


Non-Fracture Zone
Fracture Zone
Non-Fractured Zone
Density (RHOB)
Cumulative Gamma Ray (CGR)
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
−20
−10
0
10
20
30
40
50


Non-Fracture Zone
Fracture Zone
Sum of Gamma Ray (SGR)
Cumulative Gamma Ray (CGR)
140
120
100
80
60
40
20
0
−20
−20
−10
0
10
20
30
40
50


Non-Fracture Zone
Fracture Zone
Caliper (CALI)
Sonic (DT)
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
7.5
30
40
50
60
70
80
90
100


Non-Fracture Zone
Fracture Zone
Sum of Gamma Ray (SGR)
Density (DT))
140
120
100
80
60
40
20
0
−20
30
40
50
60
70
80
90
100

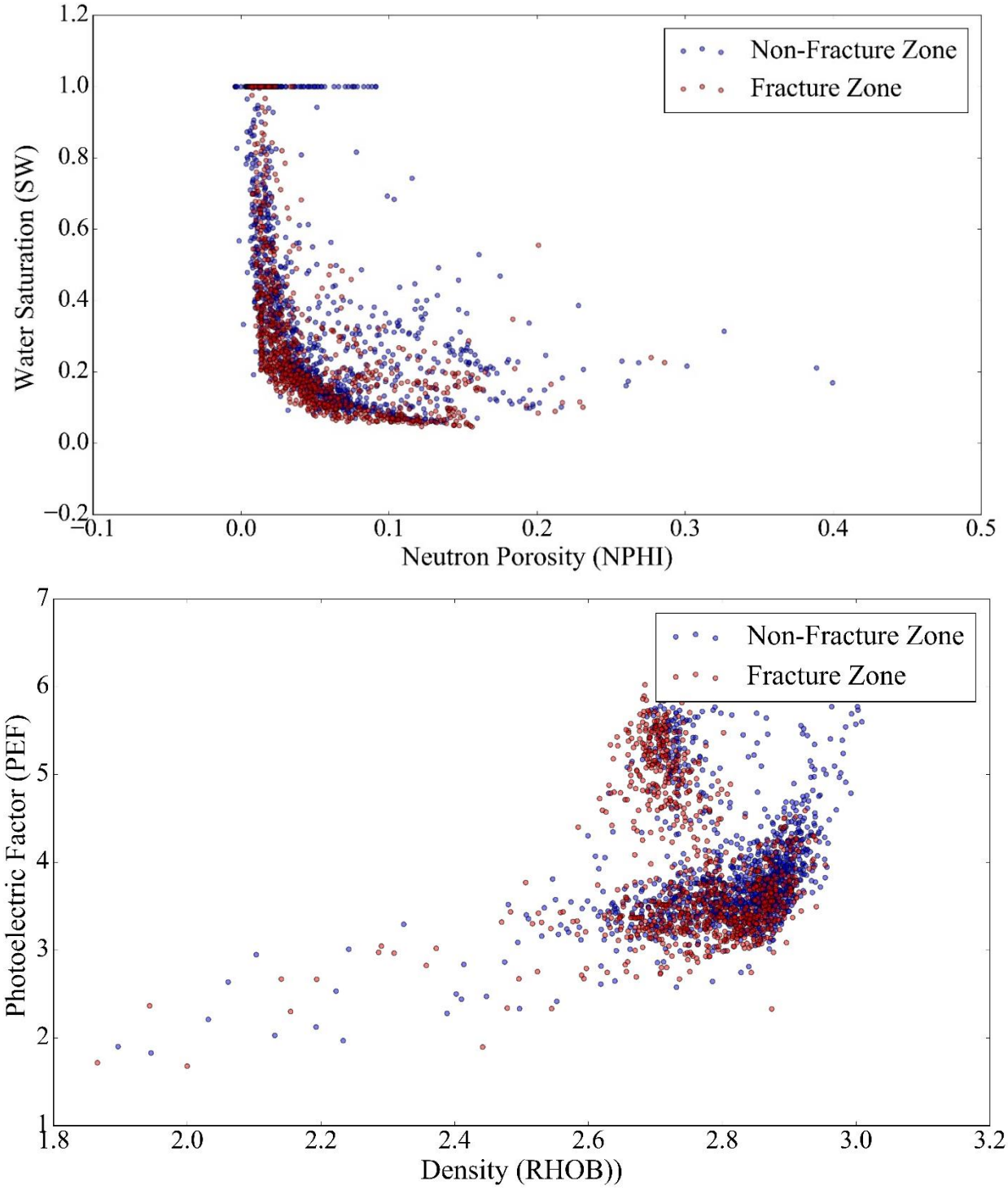
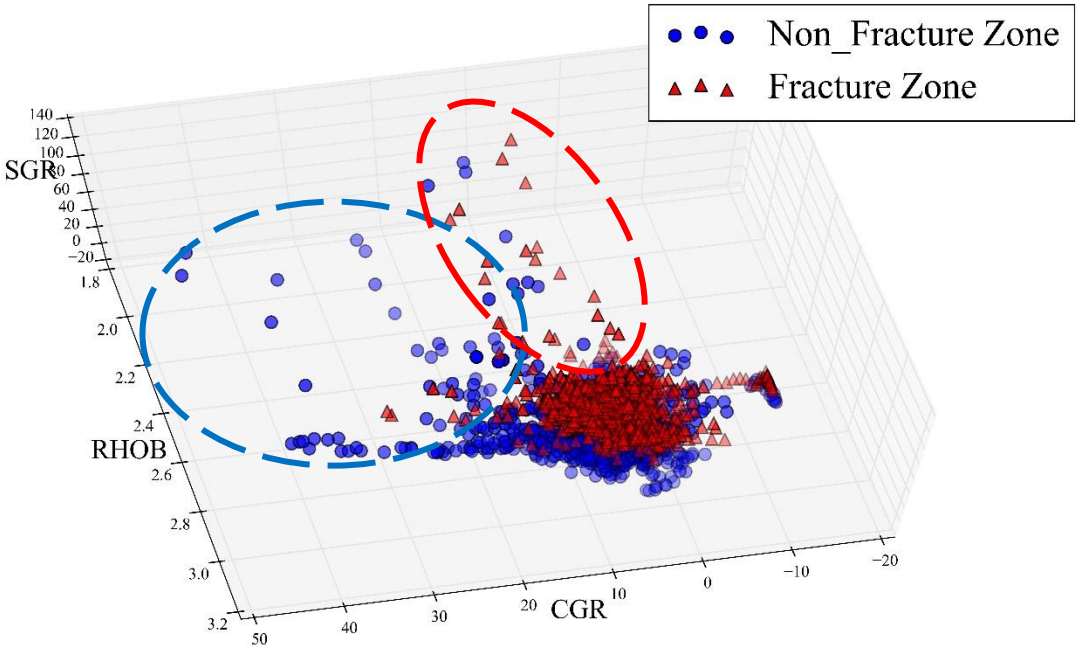
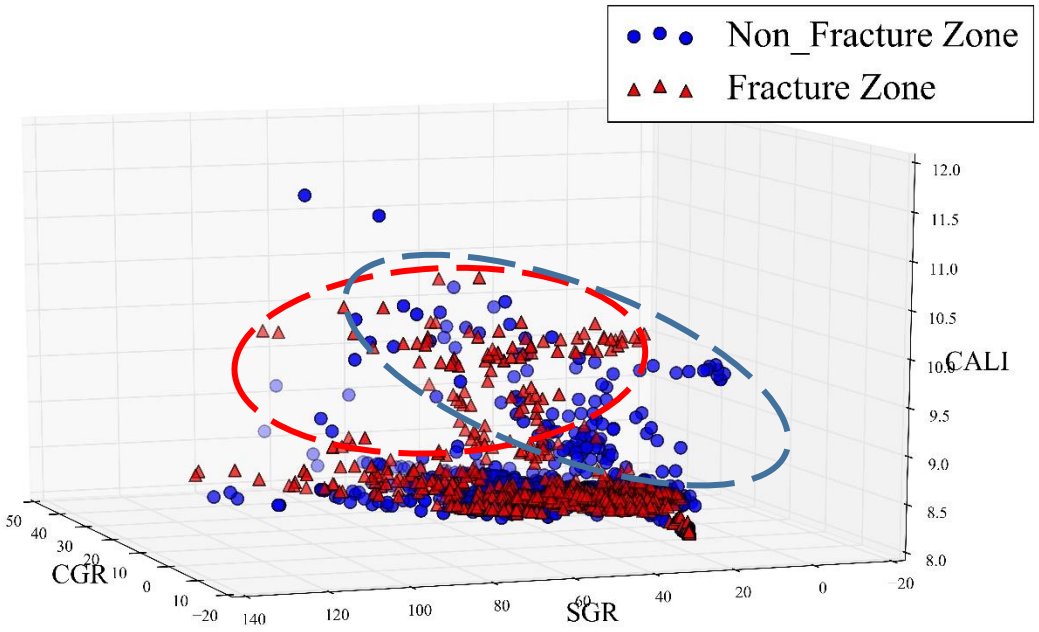


Figure 5.2 Cross plots of selected petrophysical logs in fractured and non-fractured zones in well 1.

In Figure 5.3, 3D cross plots for selected wells are displayed. Again, comparison between Figures 5.1, 5.2 and 5.3 shows that discrimination between fractured/non-fractured zones will be increased

in higher feature space. For example, in Figure 5.3, fractured zones and non-fractured ones are visually discriminated by dash lines. Three dimensional cross plots of selected logs in wells 2 and 5 are displayed in appendix K and in appendix L its code is shown.





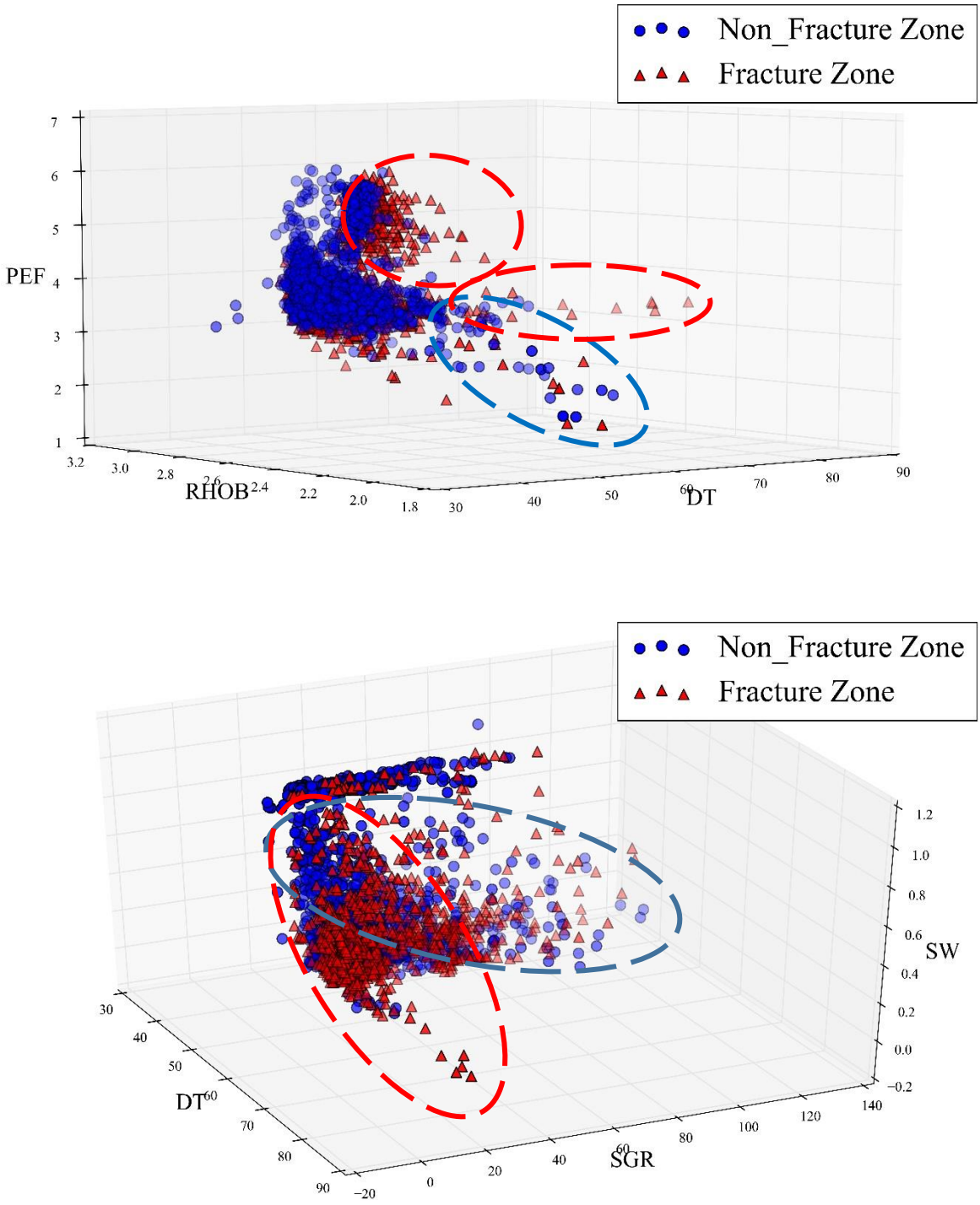


Figure 5.3 3D cross plots of selected petrophysical logs in order to show how higher feature space could help to discriminate fractured zones from non-fractured ones in well 1.

5.6 Classification of Fractured/Non-Fractured Zones

Four classifiers were applied to discriminate fractured zones from non-fractured ones by using selected petrophysical logs. Classifications were done over imbalanced and balanced data. In Table 5.4 the number of fractured/non-fractured zones in studied wells are presented.

Table 5.4 Comparison between the numbers of fractured/non-fractured zones in studied wells, and imbalance index.

Well	Non-Fractured / Fractured	Imbalance Index
1	1215 / 945	0.28
2	1608 / 1450	0.11
3	1103 / 347	2.18
4	1778 / 264	5.73
5	1056 / 690	0.53
6	1493 / 1353	0.10
7	965 / 625	0.54
8	1408 / 864	0.63

Table 5.4 obviously shows that for all wells the number of fractured zones are less than non-fractured ones. Imbalance index is defined as the difference between non-fractured and fractured

data divided by the number of fractured ones. Imbalance index approaches zero if two classes are balanced, and increases in imbalanced situations.

$$Imbalance\ Index = \frac{\text{No. of NonFractured} - \text{No. of Fractured}}{\text{No. of Fractured}} \qquad (5\text{-}1)$$

In wells 4 and 3 imbalance indexes are the highest at 5.73 and 2.18 respectively. On the opposite side, the imbalance index for wells 6 and 2 are the lowest at about 0.1. Therefore, comparison between classification in imbalanced and balanced scenarios for those four wells could be a sign of the importance of balancing. In the current study, NearMiss Under balance sampling algorithm over non-fractured zones was applied to make the balance dataset.

The four utilized classifiers are: decision tree, random forest, support vector machine and deep learning. The three first classifiers have been scripted in Jupyter iPython Notebook, and deep learning was coded in Colab.

Let's recap that CALI, CGR, RHOB, DT, PEF, NPHI, SW, PORE, PORT and SGR are utilized petrophysical logs.

To be sure that discrimination of fractured/non-fractured zones by using petrophysical logs is possible, the data of each individual well was randomly divided into training (70%) and testing (30%) sets, and classification performance over test data was investigated. It should be mentioned that fractured/non-fractured zones were labeled with 1 and zero respectively.

Confusion matrix, accuracy or correct classification rate (CCR), precision and recall have been calculated, as the classifier's performance index. Those evaluation parameters are briefly introduced as follows:

Confusion matrix (C) is a square matrix, in which entry diagonals represent the number of accurate classified data, and the rest represent misclassified data. The current study is two-class problem,

in which C_{00} represents the count of true negative, which means true classified non-fractured zones (Figure 5.4). C_{01} represents false negative or non-fractured zones which are misclassified as fractured ones. C10 represents false positive or non-fracture zones which are misclassified as fractured ones. Finally, C11 represents true positive, or the fractured zones that are correctly classified [84].

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} TN & FN \\ FP & TP \end{bmatrix}$$

Figure 5.4 Confusion Matrix.

Accuracy or Correct Classification Rate (CCR) represents the number of correctly classified data divided by the total number of the data [84]:

$$\text{Accuracy or CCR} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}} \quad (5\text{-}2)$$

where TN/TP, and FN/FP are true negative/positive, and false negative/positive respectively. Accuracy may not be a good measure if the dataset is not balanced (fractured and non-fractured classes have different number of data). In these cases, precision (positive predictive value) might be a better validation tool [84]:

$$Precision = \frac{TP}{TP + FP} \quad (5\text{-}3)$$

Obviously, one represents the highest precision and it occurs when FP is zero.

Recall is known as sensitivity or true positive rate. The highest recall is also equal to one, which means FN is zero [84]:

$$Recall = \frac{TP}{TP + FN} \quad (5\text{-}4)$$

5.6.1 Classification by Using Imbalanced Data

The results of classification over imbalanced data are presented as follows.

5.6.1.1 Decision Tree

In chapter 3, the decision tree, which is one of the most frequently and widely used supervised machine learning algorithms that can perform both regression and classification tasks, has been introduced.

In this section, the decision tree algorithm using Python's Scikit Learn library is implemented on selected petrophysical logs, in order to perform discrimination of fractured zones from non-fractured ones. Scikit-Learn contains built-in classes/methods for various decision tree algorithms (Appendix A).

In Table 5.5 the results of fractured zone detection by using decision tree classifier is presented.

Table 5.5 The results of discrimination of fractured zones from non-fractured ones using Decision Tree. Imbalanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Support
1	$\begin{bmatrix} 318 & 44 \\ 52 & 234 \end{bmatrix}$	0.85	0.82	0.84	648
2	$\begin{bmatrix} 454 & 33 \\ 53 & 378 \end{bmatrix}$	0.91	0.88	0.92	918
3	$\begin{bmatrix} 328 & 19 \\ 9 & 79 \end{bmatrix}$	0.94	0.90	0.81	435
4	$\begin{bmatrix} 532 & 10 \\ 9 & 62 \end{bmatrix}$	0.97	0.87	0.86	613
5	$\begin{bmatrix} 308 & 10 \\ 21 & 185 \end{bmatrix}$	0.94	0.90	0.95	524
6	$\begin{bmatrix} 427 & 32 \\ 22 & 373 \end{bmatrix}$	0.94	0.94	0.92	854
7	$\begin{bmatrix} 286 & 9 \\ 10 & 172 \end{bmatrix}$	0.96	0.94	0.95	477
8	$\begin{bmatrix} 414 & 0 \\ 6 & 262 \end{bmatrix}$	0.99	0.97	1	682

Surprisingly, CCR for all wells is higher than 85 percent, which much better than the results previously reported. Also, these results confirm that discrimination of fractured/non-fractured zones by using selected petrophysical logs in the studied reservoir is absolutely possible. Let's recap that, wells 3 and 4 contain the highest imbalance index (Table 5.4), and wells 2 and 6 contain the lowest index. Comparison of CCR and precision shows the highest difference for well 4. In well 4, the accuracy of classification is very high (97 %), while precision is low (87 %). This shows that the classifier is biased toward non-fractured zones, and could not recognize fractured zones. A similar problem was observed for well 3. Comparison between implantation of balanced and non-balanced data would show the importance of balancing before classification.

5.6.1.2 Random Forest

In Table 5.6 the results of classification of fractured/non-fractured zones by using random forest classifier and imbalanced data are presented. CCRs of random forest are even better than decision tree and all results are higher than 91 percent (Appendix B).

Also, the difference between accuracy and precision for wells 3 and 4 are the highest and wells 2 and 6 are the lowest. This obviously confirms the importance of balancing the data over the application of random forest.

Table 5.6 The results of discrimination of fractured zones from non-fractured ones using Random Forest. Imbalanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Support
1	$\begin{bmatrix} 342 & 20 \\ 39 & 247 \end{bmatrix}$	0.91	0.86	0.92	648
2	$\begin{bmatrix} 466 & 28 \\ 30 & 394 \end{bmatrix}$	0.94	0.93	0.93	918
3	$\begin{bmatrix} 338 & 9 \\ 15 & 73 \end{bmatrix}$	0.94	0.83	0.89	435
4	$\begin{bmatrix} 539 & 3 \\ 12 & 59 \end{bmatrix}$	0.98	0.83	0.95	613
5	$\begin{bmatrix} 309 & 9 \\ 15 & 191 \end{bmatrix}$	0.95	0.93	0.96	524
6	$\begin{bmatrix} 442 & 17 \\ 20 & 375 \end{bmatrix}$	0.96	0.95	0.96	854
7	$\begin{bmatrix} 279 & 16 \\ 9 & 173 \end{bmatrix}$	0.95	0.95	0.92	477
8	$\begin{bmatrix} 407 & 7 \\ 18 & 250 \end{bmatrix}$	0.96	0.93	0.97	682

5.6.13 Support Vector Machine

In Table 5.7 the results of support vector machine are presented. Here again CCRs are higher than 91 percent, which confirms that fractured zone detection by using petrophysical logs is possible. CCR and precision for well 4 are 95 and 78 percent respectively, which shows that in well 4, SVM is biased toward non-fractured zones, too. Another important point is that for well 8, CCR is above 99 percent, which is amazing. It seems that in this well a synthetic image log is created (Appendix D).

Table 5.7 The results of discrimination of fractured zones from non-fractured ones using Support Vector Machine. Imbalanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Support
1	$\begin{bmatrix} 311 & 33 \\ 24 & 280 \end{bmatrix}$	0.91	0.92	0.89	648
2	$\begin{bmatrix} 452 & 24 \\ 24 & 418 \end{bmatrix}$	0.95	0.94	0.94	918
3	$\begin{bmatrix} 325 & 17 \\ 10 & 83 \end{bmatrix}$	0.94	0.89	0.83	435
4	$\begin{bmatrix} 518 & 16 \\ 17 & 62 \end{bmatrix}$	0.95	0.78	0.79	613
5	$\begin{bmatrix} 303 & 9 \\ 20 & 192 \end{bmatrix}$	0.95	0.90	0.96	524
6	$\begin{bmatrix} 437 & 13 \\ 16 & 388 \end{bmatrix}$	0.97	0.96	0.97	854
7	$\begin{bmatrix} 264 & 6 \\ 10 & 197 \end{bmatrix}$	0.97	0.95	0.97	477
8	$\begin{bmatrix} 400 & 2 \\ 6 & 274 \end{bmatrix}$	0.99	0.98	0.99	682

5.6.1.4 Deep Learning

As aforementioned, for deep learning, Colab (a Google cloud-based service that replicates Jupyter Notebook in the cloud) was used. The Google Colab is free for education and research purposes and runs entirely in the cloud. TensorFlow, a free and open-source software library, was also pre-installed and optimized for the current study. Colab runs in cloud; however, one of its benefits is that it could be run over CPU to decrease CPU processing time, especially for Big Data.

In Table 5.8, the results of deep learning as well as optimum structures are represented. Here, the results are weaker than the three previous classifiers. For well 4, CCR and Precision, are both low. By better optimizing the structure and parameters of the neural network, it could be possible to achieve better results; however, this could be addressed as another shortcoming of deep learning (Appendix E).

Table 5.8 The results of discrimination of fractured zones from non-fractured ones using Deep Learning. Imbalanced version of selected petrophysical logs is used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Structure (3 layers)
1	$\begin{bmatrix}315 & 29\\ 30 & 274\end{bmatrix}$	0.91	0.90	0.90	[10 100 10]
2	$\begin{bmatrix}402 & 75\\ 61 & 380\end{bmatrix}$	0.85	0.86	0.83	[100 100 100]
3	$\begin{bmatrix}318 & 29\\ 19 & 69\end{bmatrix}$	0.89	0.78	0.70	[10 100 10]
4	$\begin{bmatrix}520 & 14\\ 19 & 61\end{bmatrix}$	0.95	0.76	0.81	[10 100 10]
5	$\begin{bmatrix}296 & 21\\ 28 & 175\end{bmatrix}$	0.90	0.86	0.89	[10 20 10]
6	$\begin{bmatrix}357 & 89\\ 90 & 311\end{bmatrix}$	0.79	0.77	0.78	[10 20 10]
7	$\begin{bmatrix}264 & 32\\ 19 & 163\end{bmatrix}$	0.89	0.89	0.83	[10 100 10]
8	$\begin{bmatrix}354 & 55\\ 75 & 193\end{bmatrix}$	0.81	0.72	0.78	[100 100 100]

5.6.1.5 Conclusion of Classification using Imbalanced Data

In Figure 5.5 CCR for different classifiers in all studied wells, as well as the average CCR are presented. Comparison between classifiers show that deep learning has the weakest discrimination, while both random forest and SVM have the best. Based on the results, random forest and SVM are the two selected classifiers for the next step of imbalanced data, which is using the database of a well to discriminate fractured zones from non-fractured ones in other wells.

Surprisingly, the best results have been for well 4, which has the highest imbalance index. This clearly shows that for an imbalanced situation, CCR might be perfect, as the aforementioned precision is dramatically low. Therefore, judgment based just on CCR could be a big mistake.

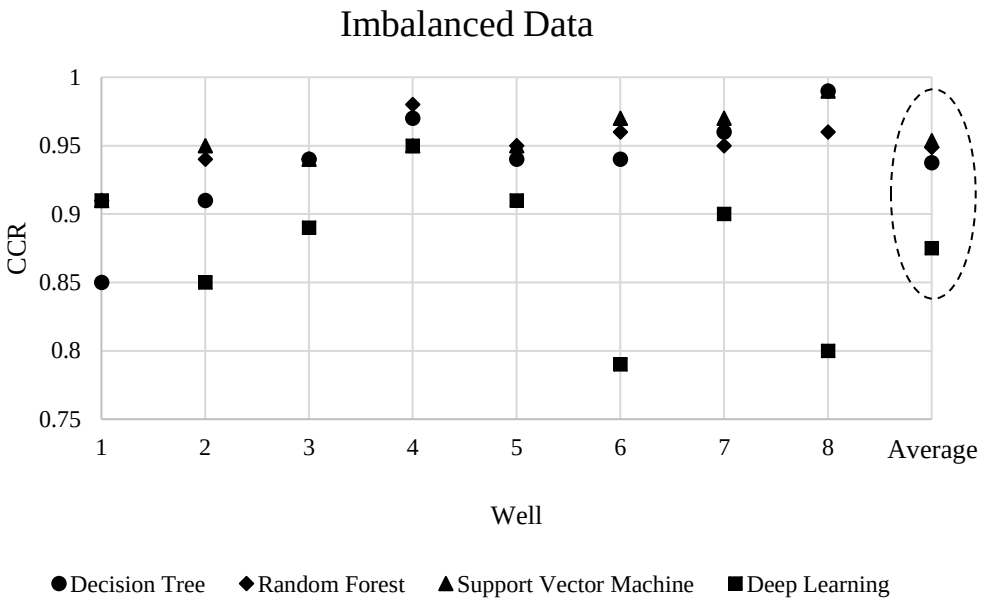


Figure 5.5 Comparison between correct classification rate (CCR) of discrimination of fractured/non-fractured zones of imbalanced data by using different classifiers over studied wells.

5.6.2 Classification by Using Balanced Data

The procedure that was applied on imbalanced data is repeated over balanced data. As aforementioned, near miss under sampling algorithm over non-fractured zones was done to make balanced data. The most focus is on the investigation of the effect of balancing over accuracy and precision of wells 3 and 4, which have the highest imbalance index. Results are presented as follows.

5.6.2.1 Decision Tree

The results of classification of fractured/non-fractured zones by using decision tree classifier and balanced petrophysical data are presented in Table 5.9. Comparison between Tables 5.5 and 5.9 shows that overall the accuracy for imbalanced data was higher than balanced data, while for wells 3 and 4, accuracy and precision in balanced data are so similar to each other. This means that in the current scenario not only non-fractured but also fractured zones are detected.

Table 5.9 The results of discrimination of fractured zones from non-fractured ones using Decision Tree. Balanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Support
1	$\begin{bmatrix} 226 & 47 \\ 57 & 237 \end{bmatrix}$	0.82	0.81	0.83	567
2	$\begin{bmatrix} 385 & 48 \\ 57 & 380 \end{bmatrix}$	0.88	0.87	0.89	870
3	$\begin{bmatrix} 101 & 9 \\ 7 & 92 \end{bmatrix}$	0.92	0.93	0.91	209
4	$\begin{bmatrix} 68 & 11 \\ 5 & 75 \end{bmatrix}$	0.90	0.94	0.87	159
5	$\begin{bmatrix} 200 & 17 \\ 14 & 183 \end{bmatrix}$	0.93	0.93	0.91	414
6	$\begin{bmatrix} 371 & 28 \\ 26 & 387 \end{bmatrix}$	0.93	0.94	0.93	812
7	$\begin{bmatrix} 177 & 12 \\ 4 & 182 \end{bmatrix}$	0.96	0.98	0.94	375
8	$\begin{bmatrix} 249 & 3 \\ 5 & 262 \end{bmatrix}$	0.98	0.98	0.99	519

5.6.2.2 Random Forest

In Table 5.10, results of random forest are reported. Both CCR and precision for all wells are higher than 88 percent, which shows capability of method in detection of fractured and non-fractured zones. The most amazing result is for well 4, where precision is even bigger than accuracy, while in the case of imbalanced data (Table 5.6), precision was 15% lower than accuracy. This might be called the magic of balancing!

Table 5.10 The results of discrimination of fractured zones from non-fractured ones using Random Forest. Balanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Support
1	$\begin{bmatrix} 237 & 36 \\ 33 & 261 \end{bmatrix}$	0.88	0.89	0.88	567
2	$\begin{bmatrix} 405 & 28 \\ 42 & 395 \end{bmatrix}$	0.92	0.90	0.93	870
3	$\begin{bmatrix} 104 & 6 \\ 7 & 92 \end{bmatrix}$	0.94	0.93	0.94	209
4	$\begin{bmatrix} 73 & 6 \\ 1 & 79 \end{bmatrix}$	0.96	0.99	0.93	159
5	$\begin{bmatrix} 198 & 19 \\ 9 & 188 \end{bmatrix}$	0.93	0.95	0.91	414
6	$\begin{bmatrix} 383 & 16 \\ 19 & 394 \end{bmatrix}$	0.96	0.95	0.96	812
7	$\begin{bmatrix} 178 & 11 \\ 4 & 182 \end{bmatrix}$	0.96	0.98	0.94	375
8	$\begin{bmatrix} 238 & 14 \\ 6 & 261 \end{bmatrix}$	0.96	0.98	0.95	519

5.6.2.3 Support Vector Machine

The results of applying SVM over balanced data are displayed in Table 5.11. Again both accuracies and precisions are great.

5.6.2.4 Deep Learning

In Table 5.12, the structure of semi-optimum network and the results of classification of fractured zones by using deep learning and balanced data are presented. Accuracies and precisions are lower than other classifiers. For sure one of reasons accuracies are not as high as other methods might be that the structure and parameters are not well optimized.

Table 5.11 The results of discrimination of fractured zones from non-fractured ones using Support Vector Machine. Balanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Support
1	$\begin{bmatrix} 253 & 24 \\ 23 & 267 \end{bmatrix}$	0.92	0.92	0.92	567
2	$\begin{bmatrix} 414 & 35 \\ 29 & 392 \end{bmatrix}$	0.93	0.93	0.92	870
3	$\begin{bmatrix} 97 & 15 \\ 10 & 87 \end{bmatrix}$	0.88	0.90	0.85	209
4	$\begin{bmatrix} 83 & 2 \\ 14 & 60 \end{bmatrix}$	0.91	0.81	0.97	159
5	$\begin{bmatrix} 182 & 32 \\ 7 & 193 \end{bmatrix}$	0.91	0.96	0.86	414
6	$\begin{bmatrix} 414 & 18 \\ 12 & 368 \end{bmatrix}$	0.96	0.97	0.95	812
7	$\begin{bmatrix} 183 & 11 \\ 6 & 175 \end{bmatrix}$	0.96	0.97	0.94	375
8	$\begin{bmatrix} 261 & 7 \\ 1 & 250 \end{bmatrix}$	0.98	1	0.97	519

Table 5.12 The results of discrimination of fractured zones from non-fractured ones using Deep Learning. Balanced version of selected petrophysical logs are used.

Well	Confusion Matrix	Accuracy or CCR	Precision	Recall	Structure (3 layers)
1	$\begin{bmatrix} 242 & 31 \\ 29 & 275 \end{bmatrix}$	0.90	0.90	0.90	[100 100 100]
2	$\begin{bmatrix} 395 & 53 \\ 41 & 380 \end{bmatrix}$	0.89	0.90	0.88	[100 100 100]
3	$\begin{bmatrix} 101 & 11 \\ 7 & 90 \end{bmatrix}$	0.91	0.93	0.89	[10 20 10]
4	$\begin{bmatrix} 86 & 1 \\ 10 & 64 \end{bmatrix}$	0.93	0.86	0.98	[10 100 10]
5	$\begin{bmatrix} 178 & 28 \\ 11 & 185 \end{bmatrix}$	0.90	0.94	0.87	[10 100 10]
6	$\begin{bmatrix} 324 & 101 \\ 98 & 294 \end{bmatrix}$	0.76	0.75	0.74	[10 20 10]
7	$\begin{bmatrix} 173 & 20 \\ 16 & 165 \end{bmatrix}$	0.90	0.91	0.89	[10 100 10]
8	$\begin{bmatrix} 211 & 57 \\ 50 & 198 \end{bmatrix}$	0.79	0.80	0.78	[10 20 10]

5.6.2.5 Conclusion of Classification using Balanced Data

In Figure 5.6 CCR for different classifiers for balanced data are presented. Similar to imbalanced data (Figure 5.5), deep learning has presented the weakest discrimination, while both random forest and SVM presented the best discrimination. Here again random forest and SVM are selected for future studies, in which classifiers will be trained by balanced data and then will be applied to discriminate fractured zones from non-fractured ones in other wells.

5.6.3 Comparison between Imbalanced and Balanced Data

In Table 5.13, average accuracy and precision of the four utilized classifiers by using imbalanced/balanced data for eight studied wells are abstracted.

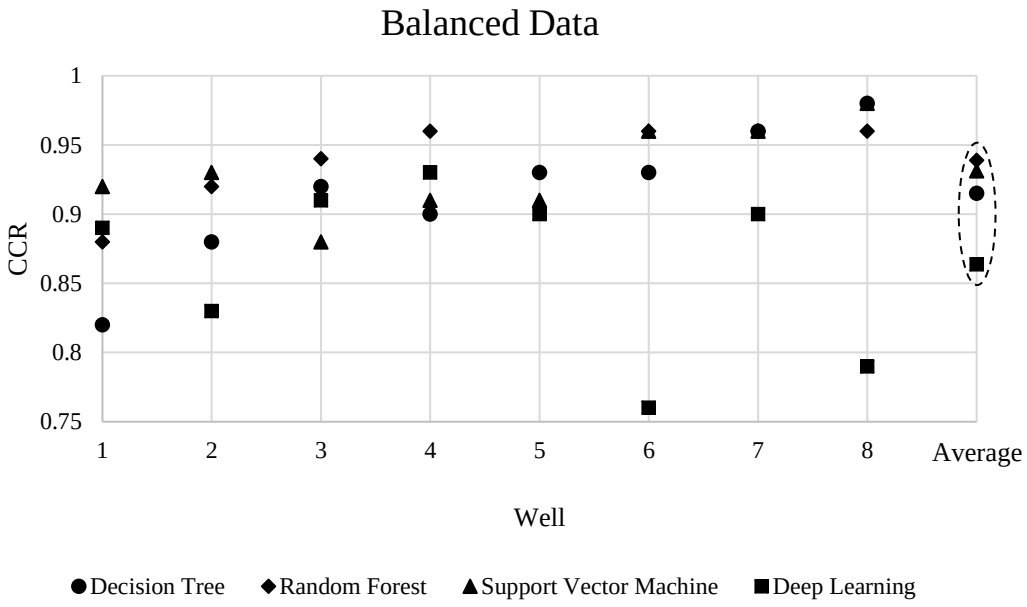


Figure 5.6. Comparison between correct classification rates (CCR) of discrimination of fractured/non-fractured zones of balanced data by using different classifiers.

Table 5.13 Comparison between average accuracy and precision of different classifiers for eight studied wells while using imbalanced and balanced data.

Well	Imbalanced Data		Balanced Data	
	Average Accuracy or CCR	Average Precision	Average Accuracy or CCR	Average Precision
1	0.895	0.875	0.880	0.880
2	0.912	0.902	0.905	0.900
3	0.927	0.850	0.912	0.922
4	0.962	0.810	0.925	0.900
5	0.935	0.897	0.917	0.945
6	0.915	0.905	0.902	0.902
7	0.942	0.932	0.945	0.960
8	0.937	0.900	0.927	0.940
Average	0.928	0.884	0.914	0.919

The difference between accuracy and precision might be an index of reliability of classification (Figure 5.7). The reason is that a low difference shows that classifiers have distinguished both fractured and non-fractured zones. As can be seen in Figure 5.7, the average difference between accuracy and precision for balanced data is almost zero, while for imbalanced data it is about 4 percent. This means that classifiers for imbalanced data were biased toward non-fractured zones, which were the dominant class. The importance of balancing is highlighted in the case of wells 3 and especially 4, where imbalance indexes were 2.18 and 5.73 respectively (Table 5.4). For example, in well 4, the difference between average accuracy and precision while using the original data is about 15 percent. After balancing, it declined to about 2 percent, amazing result!

In abstract, it might be deduced that balancing is mandatory if the imbalance index is higher than 1.

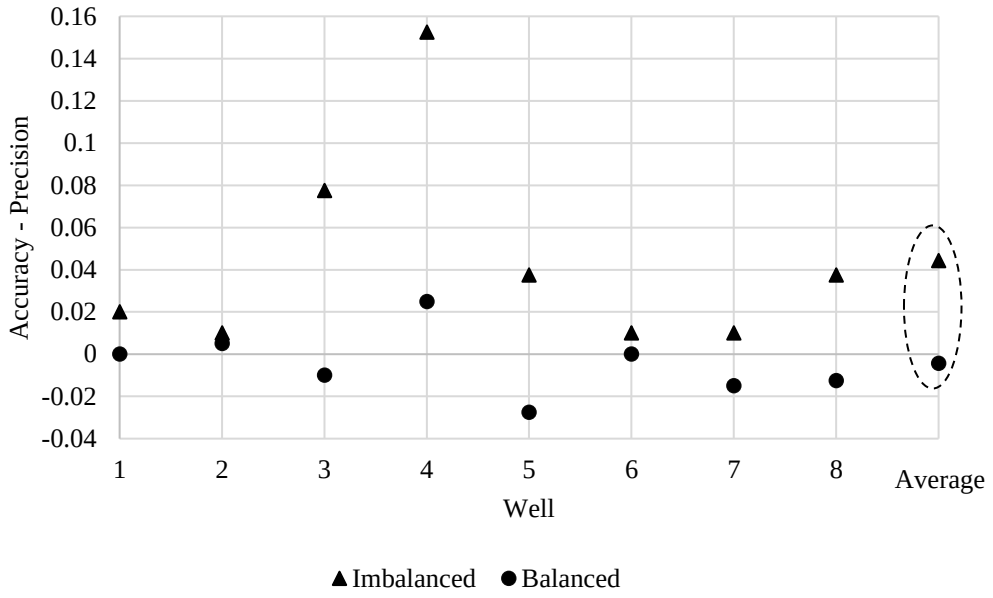


Figure 5.7. Average accuracy minus average precision of application of four utilized classifiers for balanced and imbalanced data.

5.7 Generalization of Classification

As aforementioned, in the studied oil field about 450 wells were drilled. Image and petrophysical logs were run in eight wells, which are being used in the current study, while just petrophysical logs were run in the rest. Therefore, classifiers have to be trained and utilized to discriminate fractured zones from non-fractured ones in the wells in which just petrophysical logs were run. In other words, making semi-image logs for fracture detection is the goal of this study.

In this section, one well is considered a test and the remaining seven as training wells, to check the possibility of using trained classifiers for fracture detection in other wells. Random Forest (RF) and Support Vector Machine (SVM) are the classifiers that were used. Two scenarios were considered for training wells: using an imbalanced database and balanced sets. The results are reported as follows:

5.7.1 Classification by Imbalanced Dataset

In Tables 5.14 through 5.21, confusion matrix, CCR, precision and recall resulted from classification when one well is a test and the others were utilized for training. For example, in Table 5.14, wells 2 through 8 were individually trained to discriminate fractured/non-fractured zones, and trained classifiers were used for well 1. Overall, CCR, precision and recalls are the same, which means that imbalanced data was suitable for fractured zone detection. Of course, in the case in which well 3 was the test, there are significant differences between accuracy and precision when well 4 was used as a testing set (Table 5.16). In the opposite situation (well 3: train, well 4: test, Table 5.17) a difference between accuracy and precision could also be observed. The imbalance index for wells 3 and 4 was the highest (Table 5.4). It could be interpreted that using highly imbalanced data for both training and test makes the classification biased toward the major class (non-fractured zones in the current study).

Table 5.14 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 1 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
2	1	$\begin{bmatrix}1158 & 56\\ 39 & 906\end{bmatrix}$	$\begin{bmatrix}1182 & 32\\ 38 & 907\end{bmatrix}$	0.96	0.97	0.96	0.96	0.94	0.97
3		$\begin{bmatrix}1183 & 31\\ 31 & 914\end{bmatrix}$	$\begin{bmatrix}1194 & 20\\ 37 & 908\end{bmatrix}$	0.97	0.97	0.97	0.96	0.97	0.98
4		$\begin{bmatrix}1179 & 35\\ 38 & 907\end{bmatrix}$	$\begin{bmatrix}1167 & 47\\ 33 & 912\end{bmatrix}$	0.97	0.96	0.96	0.97	0.96	0.95
5		$\begin{bmatrix}1187 & 27\\ 57 & 888\end{bmatrix}$	$\begin{bmatrix}1180 & 34\\ 39 & 906\end{bmatrix}$	0.96	0.97	0.94	0.96	0.97	0.96
6		$\begin{bmatrix}1195 & 19\\ 46 & 899\end{bmatrix}$	$\begin{bmatrix}1179 & 35\\ 39 & 906\end{bmatrix}$	0.97	0.97	0.95	0.96	0.98	0.96
7		$\begin{bmatrix}1171 & 43\\ 54 & 891\end{bmatrix}$	$\begin{bmatrix}1178 & 36\\ 33 & 912\end{bmatrix}$	0.96	0.97	0.94	0.97	0.95	0.96
8		$\begin{bmatrix}1188 & 26\\ 43 & 902\end{bmatrix}$	$\begin{bmatrix}1175 & 39\\ 39 & 906\end{bmatrix}$	0.97	0.96	0.95	0.96	0.97	0.96

Table 5.15 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 2 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	2	$\begin{bmatrix}1597 & 10\\ 38 & 1412\end{bmatrix}$	$\begin{bmatrix}1568 & 39\\ 27 & 1423\end{bmatrix}$	0.98	0.98	0.97	0.98	0.99	0.97
3		$\begin{bmatrix}1595 & 12\\ 47 & 1403\end{bmatrix}$	$\begin{bmatrix}1578 & 29\\ 25 & 1425\end{bmatrix}$	0.98	0.98	0.97	0.98	0.99	0.98
4		$\begin{bmatrix}1593 & 14\\ 41 & 1409\end{bmatrix}$	$\begin{bmatrix}1581 & 26\\ 41 & 1409\end{bmatrix}$	0.98	0.98	0.97	0.97	0.99	0.98
5		$\begin{bmatrix}1597 & 10\\ 32 & 1418\end{bmatrix}$	$\begin{bmatrix}1581 & 26\\ 38 & 1412\end{bmatrix}$	0.99	0.98	0.98	0.97	0.99	0.98
6		$\begin{bmatrix}1591 & 16\\ 42 & 1408\end{bmatrix}$	$\begin{bmatrix}1579 & 28\\ 35 & 1415\end{bmatrix}$	0.98	0.98	0.97	0.98	0.99	0.98
7		$\begin{bmatrix}1594 & 13\\ 26 & 1424\end{bmatrix}$	$\begin{bmatrix}1575 & 32\\ 35 & 1415\end{bmatrix}$	0.99	0.98	0.98	0.98	0.99	0.98
8		$\begin{bmatrix}1584 & 23\\ 31 & 1419\end{bmatrix}$	$\begin{bmatrix}1576 & 31\\ 30 & 1420\end{bmatrix}$	0.98	0.98	0.98	0.98	0.98	0.98

Table 5.16 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 3 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	3	$\begin{bmatrix}1095 & 7\\ 26 & 321\end{bmatrix}$	$\begin{bmatrix}1082 & 20\\ 14 & 333\end{bmatrix}$	0.98	0.98	0.93	0.96	0.98	0.94
2		$\begin{bmatrix}1087 & 15\\ 32 & 315\end{bmatrix}$	$\begin{bmatrix}1089 & 13\\ 11 & 336\end{bmatrix}$	0.97	0.98	0.91	0.97	0.95	0.96
4		$\begin{bmatrix}1087 & 15\\ 32 & 315\end{bmatrix}$	$\begin{bmatrix}1077 & 25\\ 28 & 319\end{bmatrix}$	0.97	0.97	0.91	0.92	0.95	0.93
5		$\begin{bmatrix}1098 & 4\\ 23 & 324\end{bmatrix}$	$\begin{bmatrix}1089 & 13\\ 13 & 334\end{bmatrix}$	0.98	0.98	0.93	0.96	0.99	0.96
6		$\begin{bmatrix}1087 & 15\\ 17 & 330\end{bmatrix}$	$\begin{bmatrix}1077 & 25\\ 16 & 331\end{bmatrix}$	0.98	0.97	0.95	0.95	0.96	0.93
7		$\begin{bmatrix}1094 & 8\\ 23 & 324\end{bmatrix}$	$\begin{bmatrix}1087 & 15\\ 15 & 332\end{bmatrix}$	0.98	0.98	0.93	0.96	0.98	0.96
8		$\begin{bmatrix}1093 & 9\\ 23 & 324\end{bmatrix}$	$\begin{bmatrix}1088 & 14\\ 12 & 335\end{bmatrix}$	0.98	0.98	0.93	0.97	0.97	0.96

Table 5.17 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 4 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	4	$\begin{bmatrix}1774 & 3\\ 5 & 259\end{bmatrix}$	$\begin{bmatrix}1756 & 21\\ 27 & 237\end{bmatrix}$	1	0.98	0.98	0.90	0.99	0.92
2		$\begin{bmatrix}1772 & 5\\ 14 & 250\end{bmatrix}$	$\begin{bmatrix}1759 & 18\\ 10 & 254\end{bmatrix}$	0.99	0.99	0.95	0.96	0.98	0.93
3		$\begin{bmatrix}1773 & 4\\ 8 & 256\end{bmatrix}$	$\begin{bmatrix}1760 & 17\\ 12 & 252\end{bmatrix}$	0.99	0.98	0.97	0.95	0.98	0.94
5		$\begin{bmatrix}1760 & 0\\ 14 & 250\end{bmatrix}$	$\begin{bmatrix}1771 & 6\\ 15 & 249\end{bmatrix}$	0.99	0.99	0.95	0.94	1	0.98
6		$\begin{bmatrix}1774 & 3\\ 8 & 256\end{bmatrix}$	$\begin{bmatrix}1762 & 15\\ 7 & 257\end{bmatrix}$	0.99	0.98	0.97	0.97	0.99	0.94
7		$\begin{bmatrix}1776 & 1\\ 16 & 248\end{bmatrix}$	$\begin{bmatrix}1763 & 14\\ 9 & 255\end{bmatrix}$	0.99	0.99	0.94	0.97	0.99	0.95
8		$\begin{bmatrix}1771 & 6\\ 10 & 254\end{bmatrix}$	$\begin{bmatrix}1767 & 10\\ 11 & 253\end{bmatrix}$	0.99	0.99	0.96	0.96	0.98	0.96

Table 5.18 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 5 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	5	$\begin{bmatrix}1045 & 10\\ 27 & 663\end{bmatrix}$	$\begin{bmatrix}1029 & 26\\ 16 & 674\end{bmatrix}$	0.98	0.98	0.96	0.98	0.99	0.96
2		$\begin{bmatrix}1039 & 16\\ 18 & 672\end{bmatrix}$	$\begin{bmatrix}1023 & 32\\ 11 & 679\end{bmatrix}$	0.98	0.98	0.97	0.98	0.98	0.95
3		$\begin{bmatrix}1049 & 6\\ 13 & 677\end{bmatrix}$	$\begin{bmatrix}1036 & 19\\ 10 & 680\end{bmatrix}$	0.98	0.98	0.98	0.99	0.99	0.97
4		$\begin{bmatrix}1037 & 18\\ 21 & 669\end{bmatrix}$	$\begin{bmatrix}1032 & 23\\ 15 & 675\end{bmatrix}$	0.98	0.98	0.97	0.98	0.97	0.97
6		$\begin{bmatrix}1041 & 14\\ 15 & 675\end{bmatrix}$	$\begin{bmatrix}1039 & 16\\ 17 & 673\end{bmatrix}$	0.98	0.98	0.98	0.98	0.98	0.98
7		$\begin{bmatrix}1042 & 13\\ 25 & 665\end{bmatrix}$	$\begin{bmatrix}1037 & 18\\ 16 & 674\end{bmatrix}$	0.98	0.98	0.96	0.98	0.98	0.97
8		$\begin{bmatrix}1037 & 18\\ 29 & 661\end{bmatrix}$	$\begin{bmatrix}1031 & 24\\ 26 & 664\end{bmatrix}$	0.97	0.97	0.96	0.96	0.97	0.97

Table 5.19 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 6 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	6	$\begin{bmatrix}1488 & 4\\ 19 & 1334\end{bmatrix}$	$\begin{bmatrix}1469 & 23\\ 15 & 1338\end{bmatrix}$	0.99	0.99	0.99	0.99	1	0.98
2		$\begin{bmatrix}1475 & 17\\ 17 & 1336\end{bmatrix}$	$\begin{bmatrix}1478 & 14\\ 19 & 1334\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
3		$\begin{bmatrix}1469 & 23\\ 34 & 1319\end{bmatrix}$	$\begin{bmatrix}1474 & 18\\ 22 & 1331\end{bmatrix}$	0.98	0.99	0.97	0.98	0.98	0.99
4		$\begin{bmatrix}1479 & 13\\ 24 & 1329\end{bmatrix}$	$\begin{bmatrix}1476 & 16\\ 17 & 1336\end{bmatrix}$	0.99	0.99	0.98	0.99	0.99	0.99
5		$\begin{bmatrix}1483 & 9\\ 24 & 1329\end{bmatrix}$	$\begin{bmatrix}1479 & 13\\ 20 & 1333\end{bmatrix}$	0.99	0.99	0.98	0.99	0.99	0.99
7		$\begin{bmatrix}1469 & 23\\ 20 & 1333\end{bmatrix}$	$\begin{bmatrix}1473 & 19\\ 34 & 1319\end{bmatrix}$	0.98	0.98	0.99	0.97	0.98	0.99
8		$\begin{bmatrix}1482 & 10\\ 21 & 1332\end{bmatrix}$	$\begin{bmatrix}1465 & 27\\ 39 & 1314\end{bmatrix}$	0.99	0.98	0.98	0.97	0.99	0.98

Table 5.20 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 7 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	7	$\begin{bmatrix}952 & 12\\ 5 & 620\end{bmatrix}$	$\begin{bmatrix}951 & 13\\ 15 & 610\end{bmatrix}$	0.99	0.98	0.99	0.98	0.98	0.98
2		$\begin{bmatrix}955 & 9\\ 8 & 617\end{bmatrix}$	$\begin{bmatrix}956 & 8\\ 9 & 616\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
3		$\begin{bmatrix}954 & 10\\ 19 & 606\end{bmatrix}$	$\begin{bmatrix}955 & 9\\ 8 & 617\end{bmatrix}$	0.98	0.99	0.97	0.99	0.98	0.99
4		$\begin{bmatrix}956 & 8\\ 8 & 617\end{bmatrix}$	$\begin{bmatrix}960 & 4\\ 18 & 607\end{bmatrix}$	0.99	0.99	0.99	0.97	0.99	0.99
5		$\begin{bmatrix}953 & 11\\ 17 & 608\end{bmatrix}$	$\begin{bmatrix}960 & 4\\ 18 & 607\end{bmatrix}$	0.98	0.99	0.97	0.97	0.98	0.99
6		$\begin{bmatrix}940 & 24\\ 12 & 613\end{bmatrix}$	$\begin{bmatrix}953 & 11\\ 24 & 601\end{bmatrix}$	0.98	0.98	0.98	0.96	0.96	0.98
8		$\begin{bmatrix}958 & 6\\ 21 & 604\end{bmatrix}$	$\begin{bmatrix}957 & 7\\ 10 & 615\end{bmatrix}$	0.98	0.99	0.97	0.98	0.99	0.99

Table 5.21 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 8 is test and imbalanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1		$\begin{bmatrix}1399 & 8\\ 12 & 852\end{bmatrix}$	$\begin{bmatrix}1400 & 7\\ 10 & 854\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
2		$\begin{bmatrix}1401 & 6\\ 10 & 854\end{bmatrix}$	$\begin{bmatrix}1398 & 9\\ 6 & 858\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
3		$\begin{bmatrix}1405 & 2\\ 5 & 859\end{bmatrix}$	$\begin{bmatrix}1398 & 9\\ 3 & 861\end{bmatrix}$	1	0.99	0.99	1	1	0.99
4	8	$\begin{bmatrix}1397 & 10\\ 11 & 853\end{bmatrix}$	$\begin{bmatrix}1402 & 5\\ 8 & 856\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
5		$\begin{bmatrix}1400 & 7\\ 15 & 849\end{bmatrix}$	$\begin{bmatrix}1397 & 10\\ 3 & 861\end{bmatrix}$	0.99	0.99	0.98	1	0.99	0.99
6		$\begin{bmatrix}1403 & 4\\ 8 & 856\end{bmatrix}$	$\begin{bmatrix}1397 & 10\\ 10 & 854\end{bmatrix}$	0.99	0.99	0.99	0.99	1	0.99
7		$\begin{bmatrix}1394 & 13\\ 13 & 851\end{bmatrix}$	$\begin{bmatrix}1405 & 2\\ 9 & 855\end{bmatrix}$	0.99	1	0.98	0.99	0.99	1

In Figure 5.8 average CCRs (ACCR) for each test well when RF or SVM were utilized for classification are displayed. The amazing point is that in all cases ACCRs are higher than 96 percent, while in previous studies, ACCRs while using Parzen, Bayesian [9] and case-based reasoning [10] classifiers were reported less than 70 percent. Therefore, RF and SVM are both great; however, the performance of RF is a little bit better than SVM (average in Figure 5.8). Surprisingly, the average accuracy for discrimination of fractured/non-fractured zones for well 4, which contain the most imbalanced data among the utilized dataset (Table 5.4), was the best. This shows that, at least in the current study, being imbalanced for test data is not important.

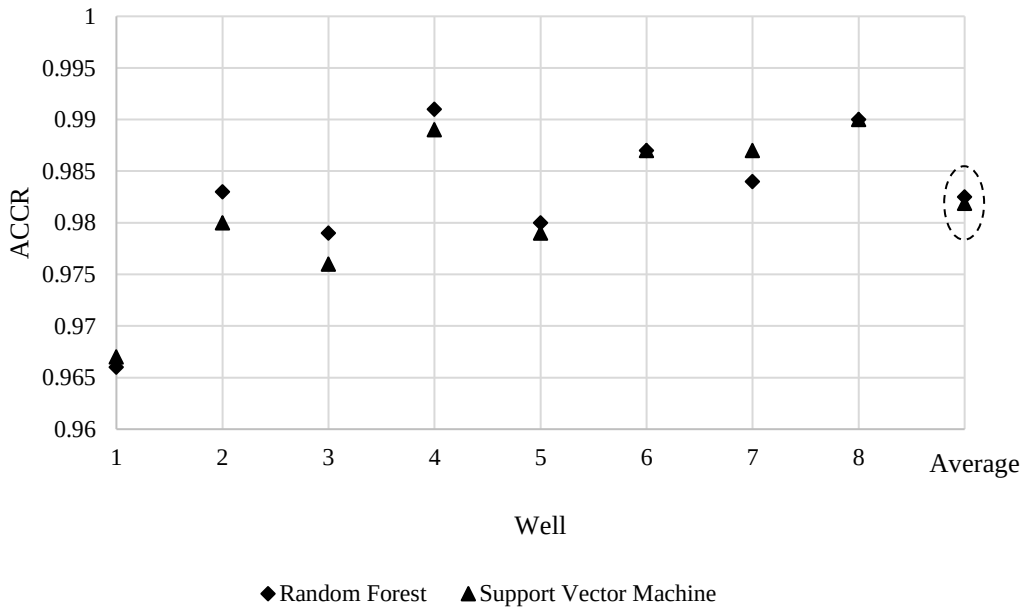


Figure 5.8. Comparison between average correct classification rate (ACCR) of discrimination of fractured/non-fractured zones by using imbalanced data resulting from different classifiers over studied wells.

In Figures 5.9 through 5.12 discrimination between fractured/non-fractured zones for wells 1 and 3, while using RF and SVM classifiers were displayed, and could be comprised by real zones achieved from the interpretation of image logs. In these figures, fractured zones are addressed by 1 and zero for non-fractured ones. All figures confirm validation of the results. However, by digging over the results, it was observed that errors occurred in the boundaries of fractured/non-fractured zones. In the industry, fractured zone recognition is much more important than resolution of the boundaries; therefore, achieved results are highly encouraging.

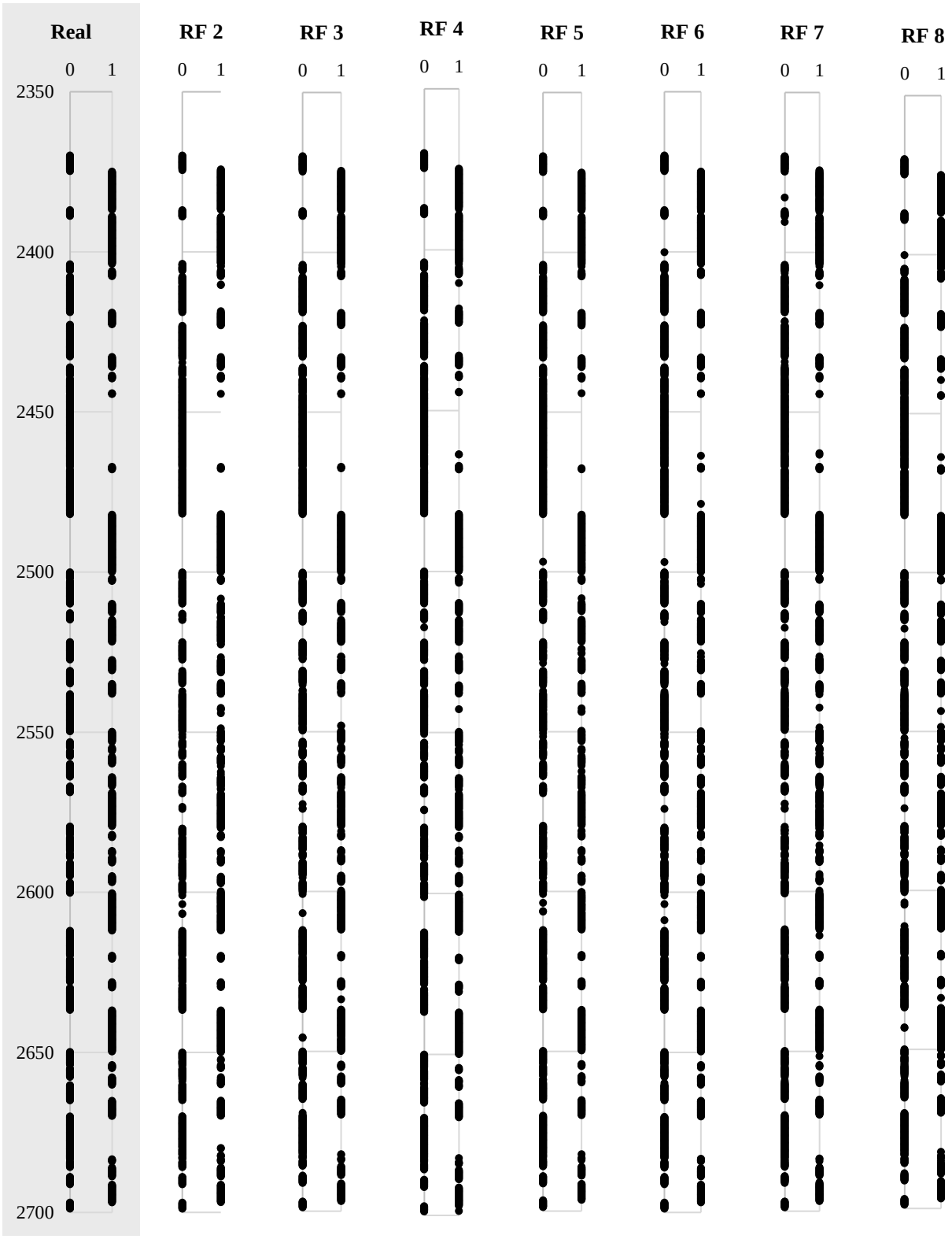


Figure 5.9. Discrimination between fractured zones and non-fractured ones for well # 1, when imbalanced version of petrophysical logs of other wells were used for training of Random Forest classifier (0: non-FZ and 1: FZ).

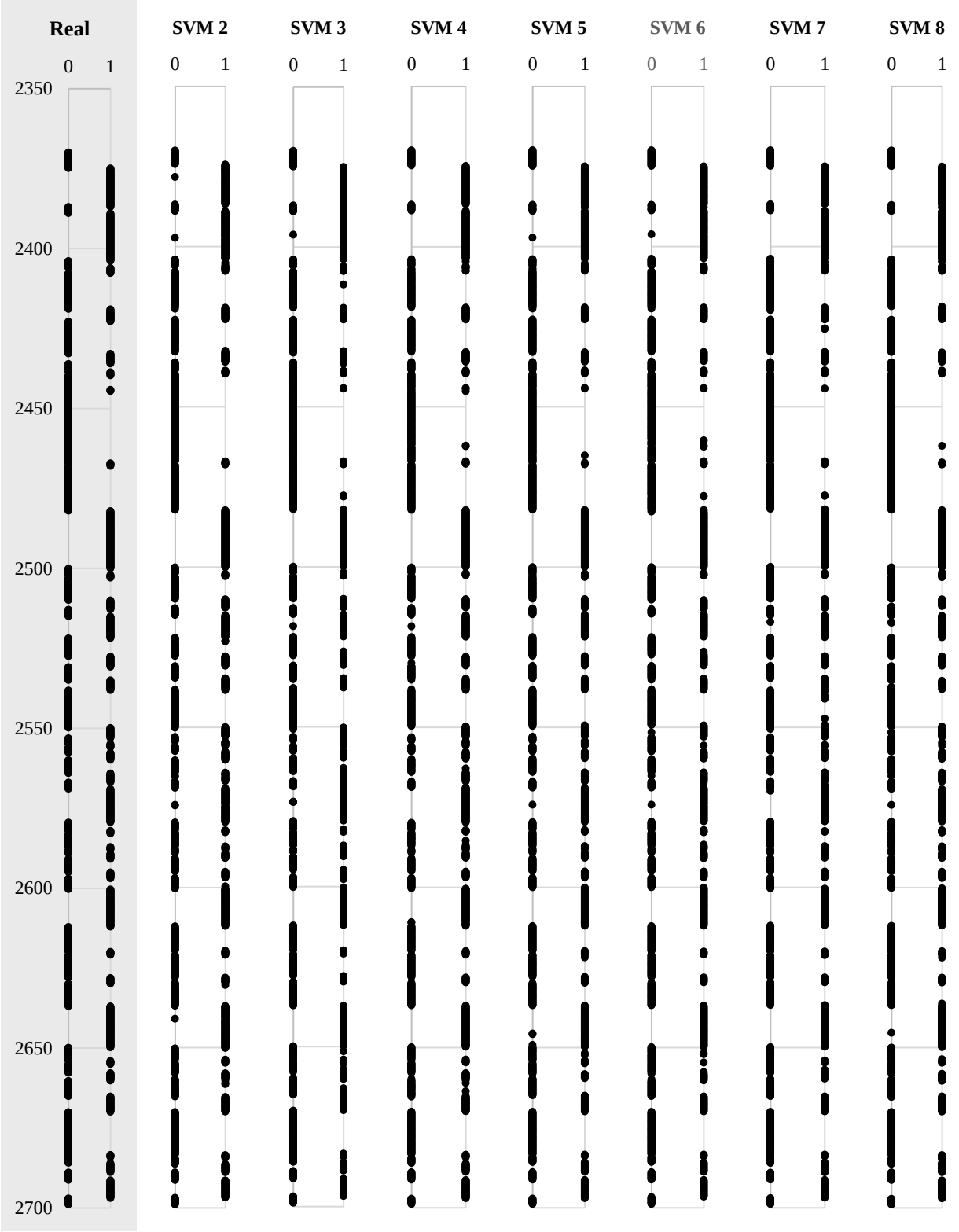


Figure 5.10. Discrimination between fractured zones and non-fractured ones for well # 1, when imbalanced version of petrophysical logs of other wells were used for training of Support Vector Machine classifier (0: non-FZ and 1: FZ).

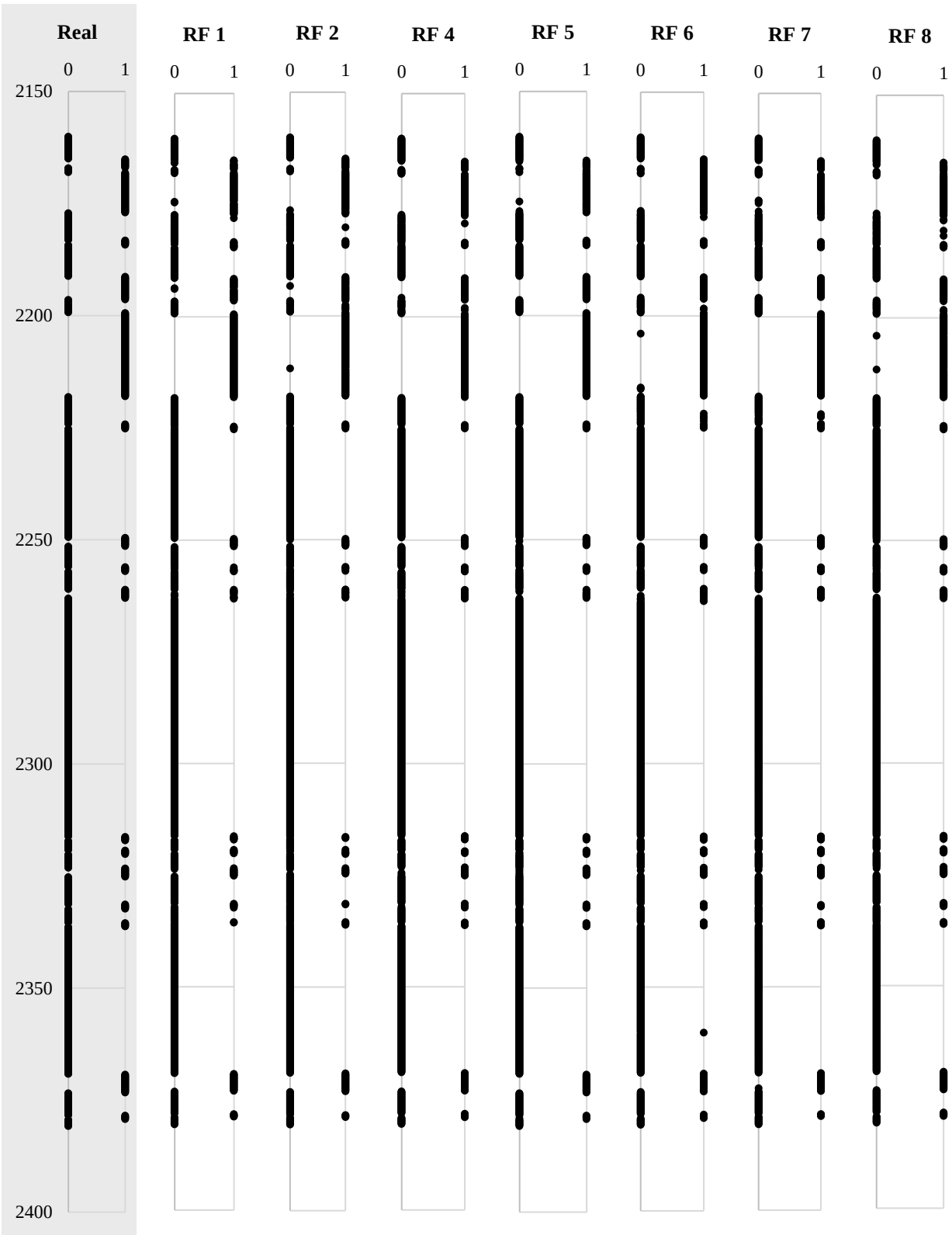


Figure 5.11. Discrimination between fractured zones and non-fractured ones for well # 3, when imbalanced version of petrophysical logs of other wells were used for training of Random Forest classifier (0: non-FZ and 1: FZ).

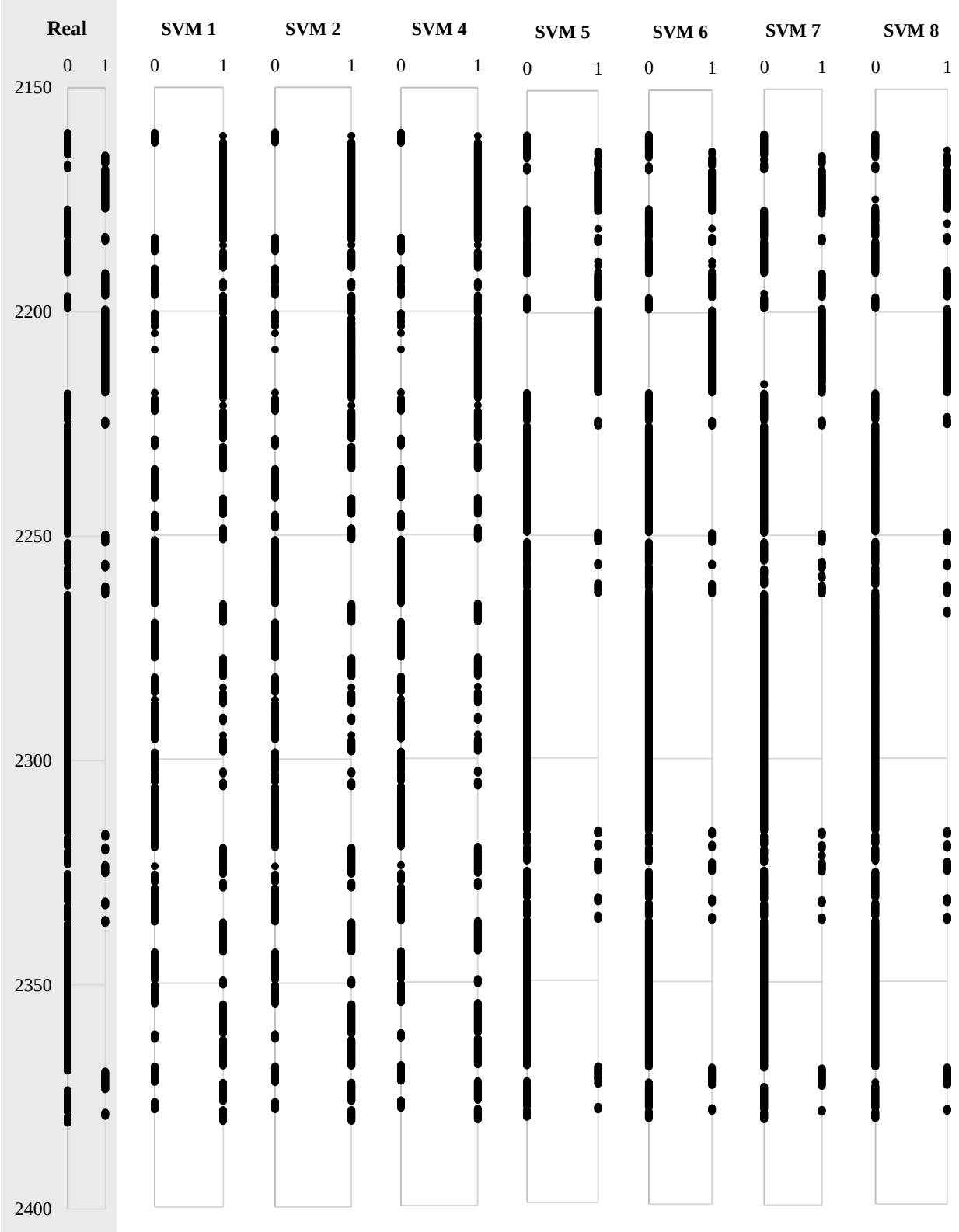


Figure 5.12. Discrimination between fractured zones and non-fractured ones for well # 3, when imbalanced version of petrophysical logs of other wells were used for training of Support Vector Machine classifier (0: non-FZ and 1: FZ).

5.7.2 Classification by Balanced Dataset

In Tables 5.22 through 5.29, the results of classification by using balanced data for training wells are presented. Like the imbalanced data, here accuracies, precisions and recalls are high and in the majority of wells are similar.

In the case that well 4 was the test well (Table 5.25), it contains the highest imbalanced dataset (Table 5.4) and the difference between accuracy and precision are high. Comparison between Tables 5.17 and 5.25 shows that no matter if training data are balanced or imbalanced, for the imbalanced test well, the difference between accuracy and precision are considerably higher than balanced test wells.

Table 5.22 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 1 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
2		$\begin{bmatrix}1176 & 38\\ 45 & 900\end{bmatrix}$	$\begin{bmatrix}1174 & 40\\ 46 & 899\end{bmatrix}$	0.96	0.96	0.95	0.95	0.96	0.96
3		$\begin{bmatrix}1189 & 25\\ 28 & 917\end{bmatrix}$	$\begin{bmatrix}1182 & 32\\ 27 & 918\end{bmatrix}$	0.98	0.97	0.97	0.97	0.97	0.97
4		$\begin{bmatrix}1186 & 28\\ 48 & 897\end{bmatrix}$	$\begin{bmatrix}1185 & 29\\ 48 & 897\end{bmatrix}$	0.96	0.96	0.95	0.95	0.97	0.97
5	1	$\begin{bmatrix}1183 & 31\\ 42 & 903\end{bmatrix}$	$\begin{bmatrix}1189 & 25\\ 43 & 902\end{bmatrix}$	0.97	0.97	0.96	0.95	0.97	0.97
6		$\begin{bmatrix}1189 & 25\\ 56 & 889\end{bmatrix}$	$\begin{bmatrix}1178 & 36\\ 40 & 905\end{bmatrix}$	0.96	0.96	0.94	0.96	0.97	0.96
7		$\begin{bmatrix}1178 & 36\\ 55 & 890\end{bmatrix}$	$\begin{bmatrix}1189 & 25\\ 33 & 912\end{bmatrix}$	0.96	0.97	0.94	0.97	0.96	0.97
8		$\begin{bmatrix}1188 & 26\\ 43 & 902\end{bmatrix}$	$\begin{bmatrix}1184 & 30\\ 47 & 898\end{bmatrix}$	0.97	0.96	0.95	0.95	0.97	0.97

Table 5.23 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 2 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	2	$\begin{bmatrix}1584 & 23\\ 34 & 1416\end{bmatrix}$	$\begin{bmatrix}1578 & 29\\ 32 & 1418\end{bmatrix}$	0.98	0.98	0.98	0.98	0.98	0.98
3		$\begin{bmatrix}1588 & 19\\ 35 & 1415\end{bmatrix}$	$\begin{bmatrix}1587 & 20\\ 21 & 1429\end{bmatrix}$	0.98	0.99	0.98	0.98	0.99	0.99
4		$\begin{bmatrix}1586 & 21\\ 39 & 1411\end{bmatrix}$	$\begin{bmatrix}1573 & 34\\ 36 & 1414\end{bmatrix}$	0.98	0.98	0.97	0.98	0.99	0.98
5		$\begin{bmatrix}1584 & 23\\ 40 & 1410\end{bmatrix}$	$\begin{bmatrix}1575 & 32\\ 28 & 1422\end{bmatrix}$	0.98	0.98	0.97	0.98	0.98	0.98
6		$\begin{bmatrix}1574 & 33\\ 41 & 1409\end{bmatrix}$	$\begin{bmatrix}1557 & 50\\ 65 & 1385\end{bmatrix}$	0.98	0.97	0.97	0.96	0.98	0.97
7		$\begin{bmatrix}1587 & 20\\ 39 & 1411\end{bmatrix}$	$\begin{bmatrix}1575 & 33\\ 31 & 1419\end{bmatrix}$	0.98	0.98	0.97	0.98	0.99	0.98
8		$\begin{bmatrix}1589 & 18\\ 41 & 1409\end{bmatrix}$	$\begin{bmatrix}1566 & 41\\ 62 & 1388\end{bmatrix}$	0.98	0.97	0.97	0.96	0.99	0.97

Table 5.24 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 3 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	3	$\begin{bmatrix}1096 & 6\\ 26 & 321\end{bmatrix}$	$\begin{bmatrix}1084 & 18\\ 14 & 333\end{bmatrix}$	0.98	0.98	0.98	0.98	0.98	0.98
2		$\begin{bmatrix}1092 & 10\\ 14 & 333\end{bmatrix}$	$\begin{bmatrix}1087 & 15\\ 9 & 338\end{bmatrix}$	0.98	0.98	0.98	0.99	0.99	0.99
4		$\begin{bmatrix}1092 & 10\\ 14 & 333\end{bmatrix}$	$\begin{bmatrix}1090 & 12\\ 9 & 338\end{bmatrix}$	0.98	0.99	0.97	0.98	0.99	0.98
5		$\begin{bmatrix}1098 & 4\\ 22 & 325\end{bmatrix}$	$\begin{bmatrix}1092 & 10\\ 17 & 330\end{bmatrix}$	0.98	0.98	0.97	0.98	0.98	0.98
6		$\begin{bmatrix}1088 & 14\\ 18 & 329\end{bmatrix}$	$\begin{bmatrix}1077 & 25\\ 16 & 331\end{bmatrix}$	0.98	0.97	0.97	0.96	0.98	0.97
7		$\begin{bmatrix}1084 & 18\\ 18 & 329\end{bmatrix}$	$\begin{bmatrix}1071 & 31\\ 16 & 331\end{bmatrix}$	0.98	0.97	0.97	0.98	0.99	0.98
8		$\begin{bmatrix}1091 & 11\\ 23 & 324\end{bmatrix}$	$\begin{bmatrix}1091 & 11\\ 15 & 332\end{bmatrix}$	0.98	0.98	0.97	0.96	0.99	0.97

Table 5.25 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 4 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	4	$\begin{bmatrix}1772 & 5\\ 11 & 253\end{bmatrix}$	$\begin{bmatrix}1763 & 14\\ 7 & 257\end{bmatrix}$	0.99	0.99	0.95	0.97	0.98	0.95
2		$\begin{bmatrix}1776 & 1\\ 14 & 250\end{bmatrix}$	$\begin{bmatrix}1765 & 12\\ 8 & 256\end{bmatrix}$	0.99	0.99	0.95	0.97	1	0.96
3		$\begin{bmatrix}1772 & 45\\ 11 & 253\end{bmatrix}$	$\begin{bmatrix}1754 & 23\\ 30 & 234\end{bmatrix}$	0.97	0.97	0.96	0.89	0.85	0.91
5		$\begin{bmatrix}1775 & 2\\ 17 & 247\end{bmatrix}$	$\begin{bmatrix}1763 & 14\\ 13 & 251\end{bmatrix}$	0.99	0.99	0.94	0.95	0.99	0.95
6		$\begin{bmatrix}1771 & 6\\ 23 & 241\end{bmatrix}$	$\begin{bmatrix}1757 & 20\\ 14 & 250\end{bmatrix}$	0.99	0.98	0.91	0.95	0.98	0.93
7		$\begin{bmatrix}1775 & 2\\ 14 & 250\end{bmatrix}$	$\begin{bmatrix}1763 & 14\\ 14 & 250\end{bmatrix}$	0.99	0.99	0.95	0.95	0.99	0.95
8		$\begin{bmatrix}1774 & 3\\ 23 & 241\end{bmatrix}$	$\begin{bmatrix}1763 & 14\\ 10 & 254\end{bmatrix}$	0.99	0.99	0.91	0.96	0.99	0.95

Table 5.26 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 5 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	5	$\begin{bmatrix}1042 & 13\\ 22 & 668\end{bmatrix}$	$\begin{bmatrix}1031 & 24\\ 22 & 668\end{bmatrix}$	0.98	0.97	0.97	0.97	0.98	0.97
2		$\begin{bmatrix}1048 & 7\\ 12 & 678\end{bmatrix}$	$\begin{bmatrix}1038 & 17\\ 16 & 674\end{bmatrix}$	0.99	0.98	0.98	0.98	0.99	0.98
3		$\begin{bmatrix}1043 & 12\\ 15 & 675\end{bmatrix}$	$\begin{bmatrix}1034 & 21\\ 30 & 660\end{bmatrix}$	0.98	0.97	0.98	0.96	0.98	0.97
4		$\begin{bmatrix}1036 & 19\\ 16 & 674\end{bmatrix}$	$\begin{bmatrix}1030 & 25\\ 21 & 669\end{bmatrix}$	0.98	0.97	0.98	0.97	0.97	0.96
6		$\begin{bmatrix}1043 & 12\\ 26 & 664\end{bmatrix}$	$\begin{bmatrix}1025 & 30\\ 15 & 675\end{bmatrix}$	0.98	0.97	0.96	0.98	0.98	0.96
7		$\begin{bmatrix}1036 & 19\\ 27 & 663\end{bmatrix}$	$\begin{bmatrix}1038 & 17\\ 21 & 669\end{bmatrix}$	0.97	0.98	0.96	0.97	0.97	0.98
8		$\begin{bmatrix}1038 & 17\\ 5 & 685\end{bmatrix}$	$\begin{bmatrix}1030 & 25\\ 19 & 671\end{bmatrix}$	0.99	0.97	0.99	0.97	0.98	0.96

Table 5.27 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 6 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	6	$\begin{bmatrix}1479 & 13\\ 20 & 1333\end{bmatrix}$	$\begin{bmatrix}1474 & 18\\ 19 & 1334\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
2		$\begin{bmatrix}1476 & 16\\ 25 & 1328\end{bmatrix}$	$\begin{bmatrix}1474 & 18\\ 21 & 1332\end{bmatrix}$	0.99	0.99	0.98	0.98	0.99	0.99
3		$\begin{bmatrix}1471 & 21\\ 18 & 1335\end{bmatrix}$	$\begin{bmatrix}1483 & 9\\ 20 & 1333\end{bmatrix}$	0.99	0.99	0.99	0.99	0.98	0.99
4		$\begin{bmatrix}1476 & 16\\ 22 & 1331\end{bmatrix}$	$\begin{bmatrix}1459 & 33\\ 12 & 1341\end{bmatrix}$	0.99	0.98	0.98	0.99	0.99	0.98
5		$\begin{bmatrix}1480 & 12\\ 22 & 1331\end{bmatrix}$	$\begin{bmatrix}1474 & 18\\ 37 & 1316\end{bmatrix}$	0.99	0.98	0.98	0.97	0.99	0.99
7		$\begin{bmatrix}1478 & 14\\ 21 & 1332\end{bmatrix}$	$\begin{bmatrix}1480 & 12\\ 18 & 1335\end{bmatrix}$	0.99	0.99	0.98	0.99	0.99	0.99
8		$\begin{bmatrix}1472 & 20\\ 26 & 1327\end{bmatrix}$	$\begin{bmatrix}1477 & 15\\ 17 & 1336\end{bmatrix}$	0.98	0.99	0.98	0.99	0.99	0.99

Table 5.28 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 7 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	7	$\begin{bmatrix}954 & 10\\ 12 & 613\end{bmatrix}$	$\begin{bmatrix}956 & 8\\ 7 & 618\end{bmatrix}$	0.99	0.99	0.98	0.99	0.98	0.99
2		$\begin{bmatrix}957 & 7\\ 4 & 621\end{bmatrix}$	$\begin{bmatrix}955 & 9\\ 9 & 616\end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
3		$\begin{bmatrix}949 & 15\\ 10 & 615\end{bmatrix}$	$\begin{bmatrix}944 & 20\\ 9 & 616\end{bmatrix}$	0.98	0.98	0.98	0.99	0.98	0.97
4		$\begin{bmatrix}958 & 6\\ 21 & 604\end{bmatrix}$	$\begin{bmatrix}938 & 26\\ 20 & 605\end{bmatrix}$	0.98	0.97	0.97	0.97	0.99	0.96
5		$\begin{bmatrix}954 & 10\\ 16 & 609\end{bmatrix}$	$\begin{bmatrix}938 & 26\\ 20 & 605\end{bmatrix}$	0.98	0.97	0.97	0.97	0.98	0.96
6		$\begin{bmatrix}955 & 9\\ 17 & 608\end{bmatrix}$	$\begin{bmatrix}950 & 14\\ 14 & 611\end{bmatrix}$	0.98	0.98	0.97	0.98	0.99	0.98
8		$\begin{bmatrix}946 & 18\\ 13 & 612\end{bmatrix}$	$\begin{bmatrix}953 & 11\\ 10 & 615\end{bmatrix}$	0.98	0.99	0.98	0.98	0.97	0.98

Table 5.29 Results of discrimination between fractured/non-fractured zones using RF and SVM when well 8 is test and balanced data of other wells are train.

Training Wells	Test Well	Confusion Matrix		Accuracy or CCR		Precision		Recall	
		RF	SVM	RF	SVM	RF	SVM	RF	SVM
1	8	$\begin{bmatrix} 1395 & 12 \\ 4 & 860 \end{bmatrix}$	$\begin{bmatrix} 1403 & 4 \\ 9 & 855 \end{bmatrix}$	0.99	0.99	1	0.99	0.99	1
2		$\begin{bmatrix} 1406 & 1 \\ 1 & 863 \end{bmatrix}$	$\begin{bmatrix} 1403 & 4 \\ 4 & 860 \end{bmatrix}$	1	1	1	1	1	1
3		$\begin{bmatrix} 1404 & 3 \\ 7 & 857 \end{bmatrix}$	$\begin{bmatrix} 1404 & 3 \\ 2 & 862 \end{bmatrix}$	1	1	0.99	1	1	1
4		$\begin{bmatrix} 1399 & 8 \\ 8 & 856 \end{bmatrix}$	$\begin{bmatrix} 1402 & 5 \\ 8 & 856 \end{bmatrix}$	0.99	0.99	0.99	0.99	0.99	0.99
5		$\begin{bmatrix} 1398 & 9 \\ 7 & 857 \end{bmatrix}$	$\begin{bmatrix} 1395 & 12 \\ 3 & 861 \end{bmatrix}$	0.99	0.99	0.99	1	0.99	0.99
6		$\begin{bmatrix} 1391 & 16 \\ 10 & 854 \end{bmatrix}$	$\begin{bmatrix} 1406 & 1 \\ 14 & 850 \end{bmatrix}$	0.99	0.99	0.99	0.98	0.98	1
7		$\begin{bmatrix} 1393 & 14 \\ 8 & 856 \end{bmatrix}$	$\begin{bmatrix} 1402 & 5 \\ 4 & 860 \end{bmatrix}$	0.99	1	0.99	1	0.98	0.99

In Figure 5.13, average CCRs (ACCR) for each test well while using balanced data for training are displayed. Here, similar to using imbalanced data for training (Figure 5.8), all ACCRs are higher than 96 percent. Therefore, no matter if balanced or imbalanced data were used for training, discrimination between fractured/non-fractured zones are acceptable. Here again, the average accuracy for RF is a little bit better than SVM (average in Figure 5.13).

Comparison between Figures 5.8 and 5.13 shows that the average accuracy for well 4 using imbalanced data is better than using balanced data. Therefore, based on the studied case, in imbalanced wells, imbalanced data are welcome.

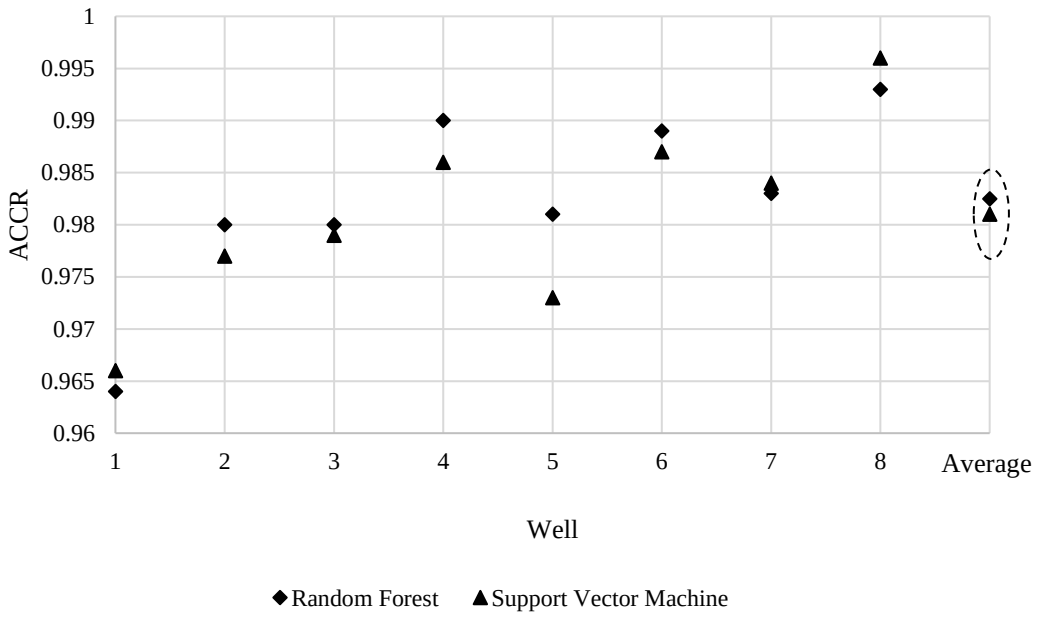


Figure 5.13. Comparison between average correct classification rate (ACCR) of discrimination of fractured/non-fractured zones by using balanced data resulting from different classifiers over studied wells.

In Figures 5.14 through 5.17, discrimination between fractured/non-fractured zones for wells 1 and 3 using balanced data for training wells is presented. Here again all figures confirm validation of the results, and digging over the results, also showed that errors occurred in the boundaries of fractured/non-fractured zones.

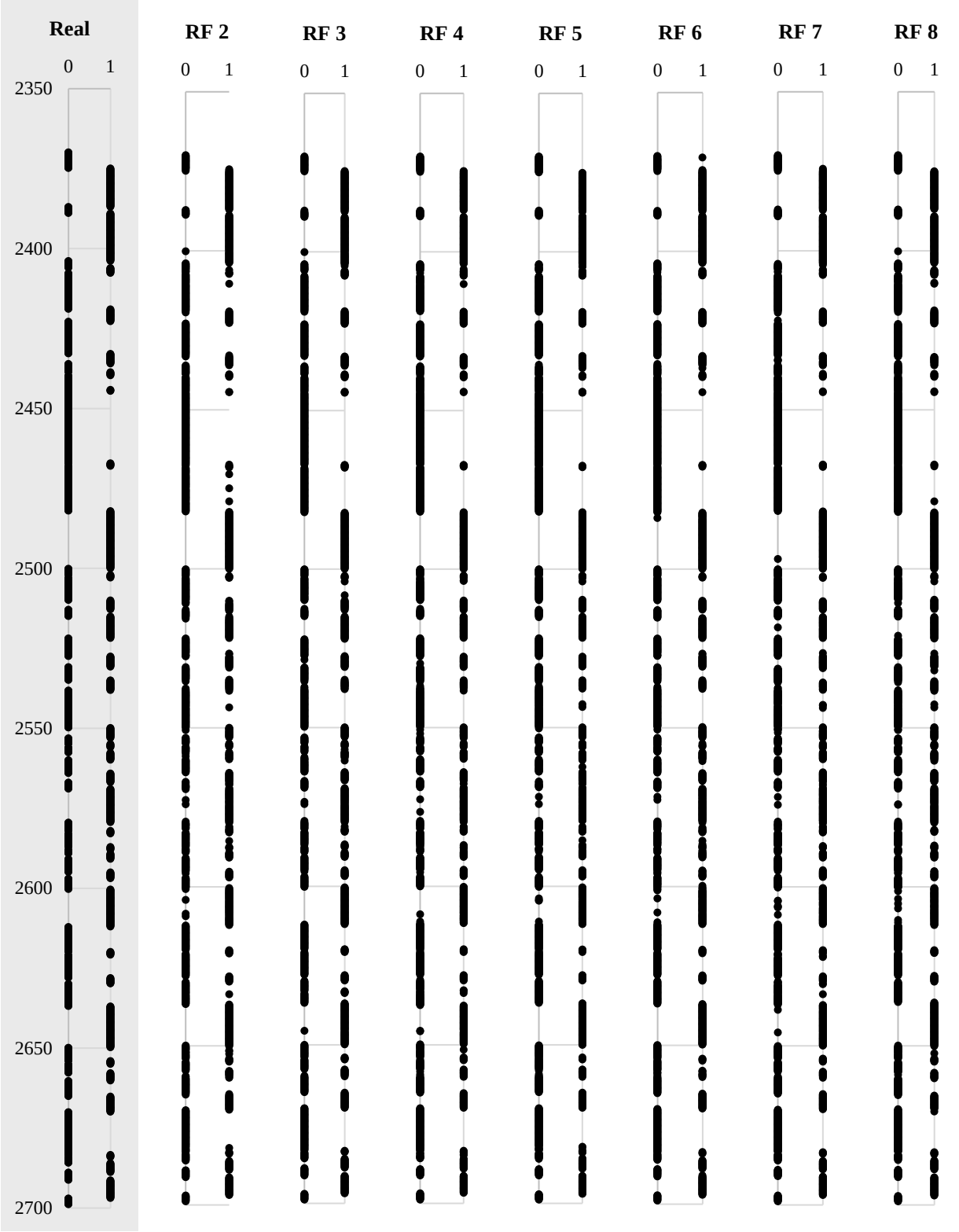


Figure 5.14. Discrimination between fractured zones and non-fractured ones for well # 1, when balanced version of petrophysical logs of other wells were used for training of Random Forest classifier (0: non-FZ and 1: FZ).

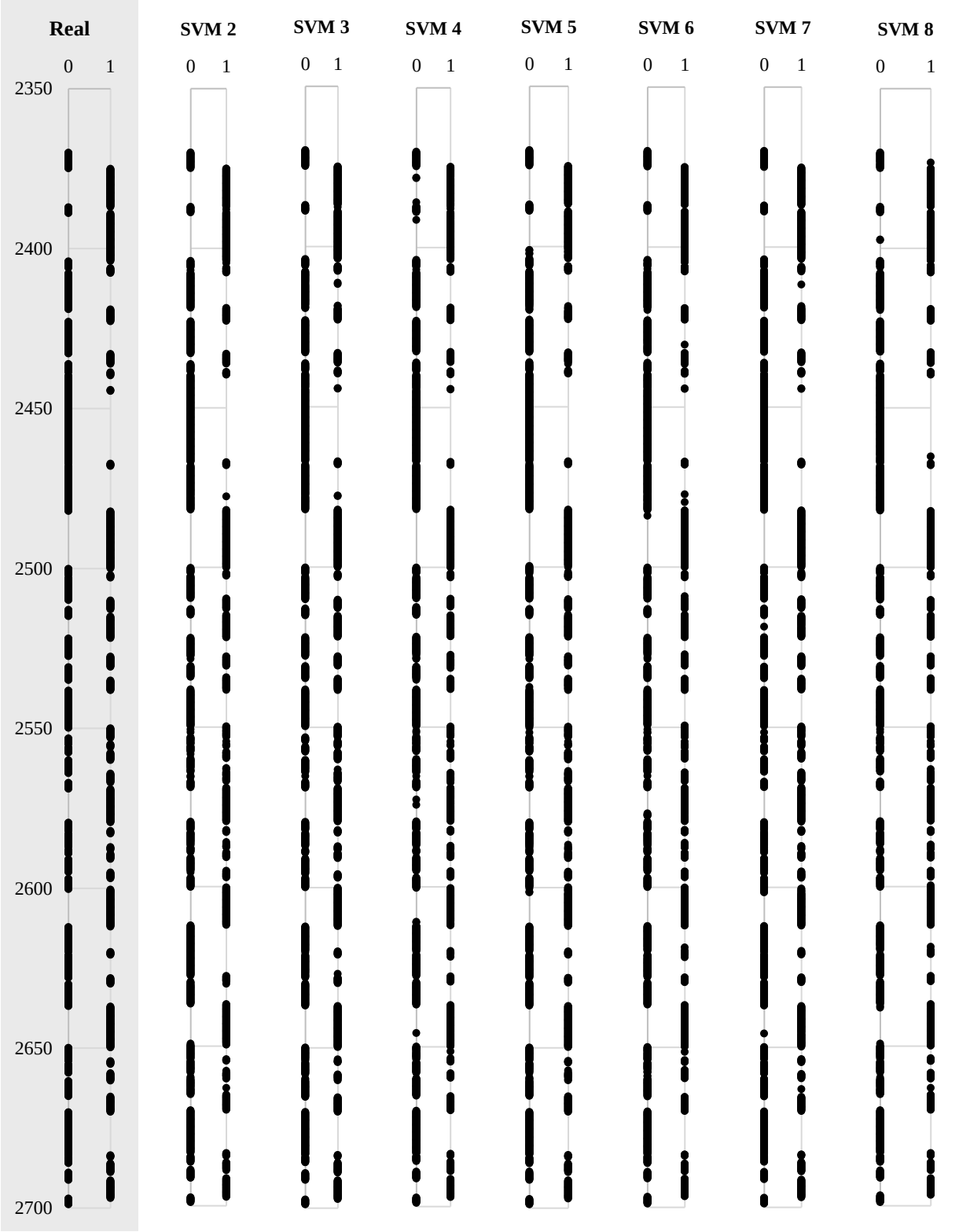


Figure 5.15. Discrimination between fractured zones and non-fractured ones for well # 1, when balanced version of petrophysical logs of other wells were used for training of Support Vector Machine classifier (0: non-FZ and 1: FZ).

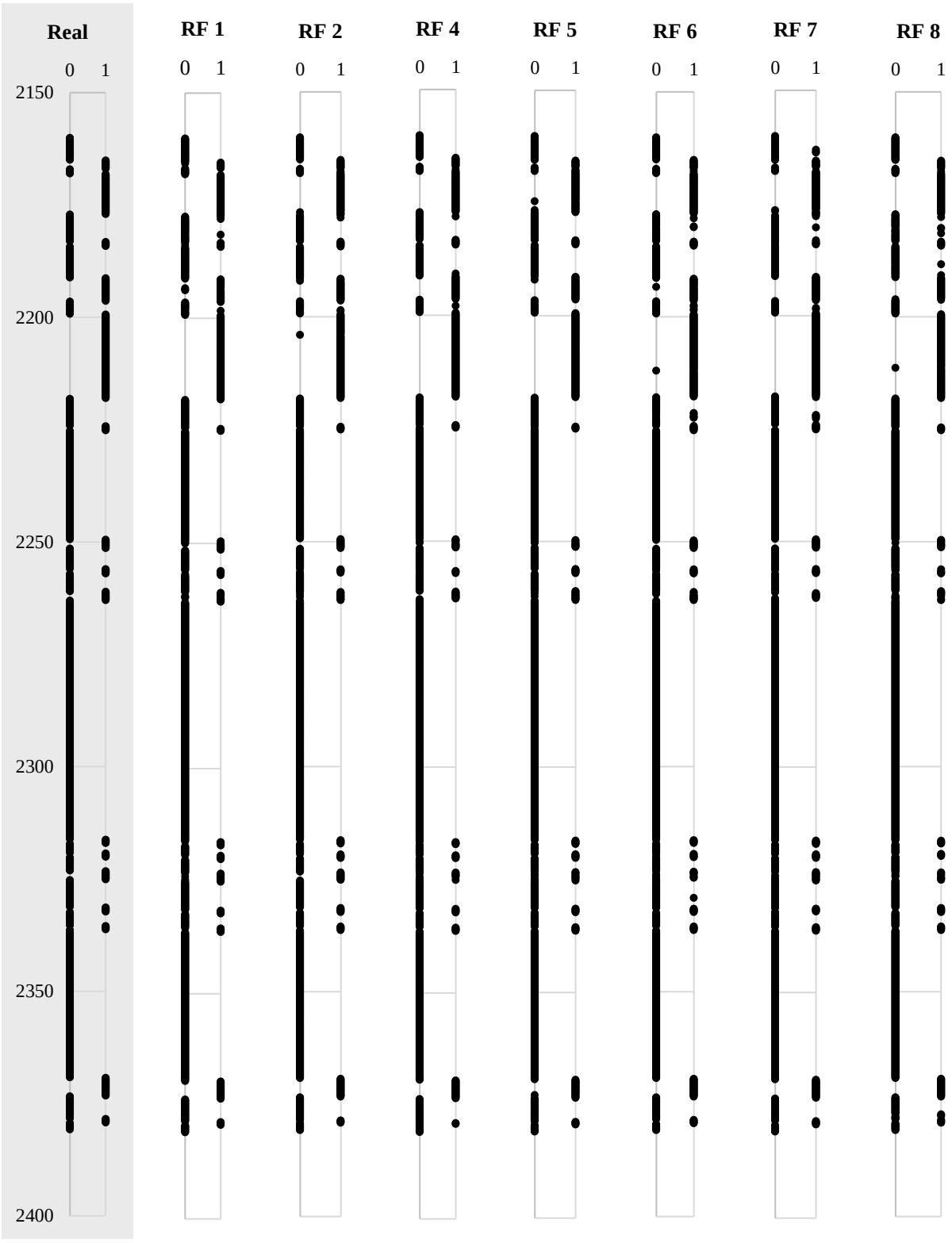


Figure 5.16. Discrimination between fractured zones and non-fractured ones for well # 3, when balanced version of petrophysical logs of other wells were used for training of Random Forest classifier (0: non-FZ and 1: FZ).

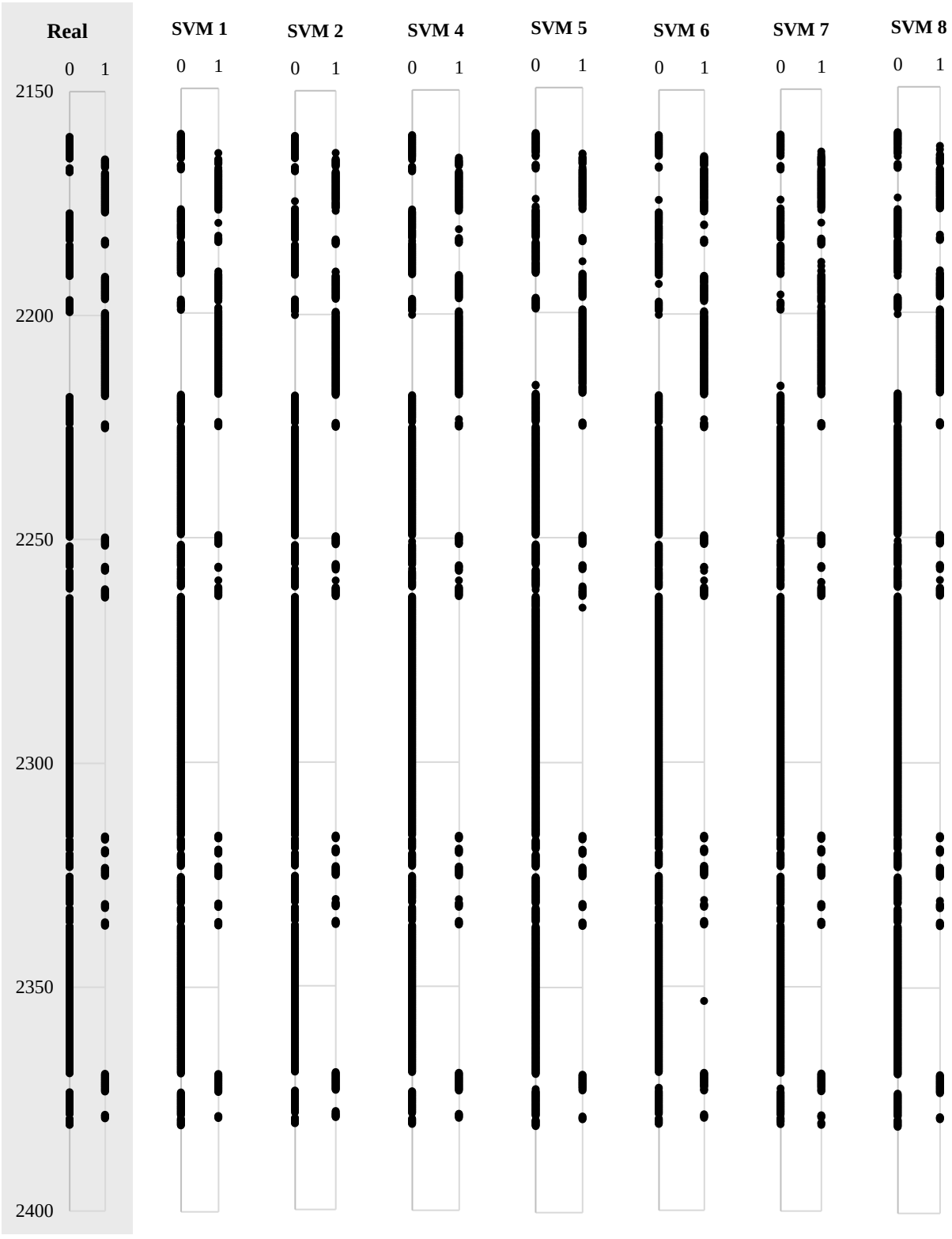


Figure 5.17. Discrimination between fractured zones and non-fractured ones for well # 3, when balanced version of petrophysical logs of other wells were used for training of Support Vector Machine classifier (0: non-FZ and 1: FZ).

5.7.3 Comparison between Balanced/Imbalanced Dataset

In Figures 5.8 and 5.13, average CCRs of fracture zone classification by RF and SVM, while one of the wells were test and the other wells were training, are presented. In the mentioned figures, imbalanced and balanced data were utilized for training wells respectively. Here, average CCRs achieved from both RF and SVM classifiers are calculated, and results are reported (Figure 5.18). Based on Figure 5.18, there are not considerable differences between the results of balanced and imbalanced data. Therefore, it could be concluded that classification is not sensitive to the balancing of training data.

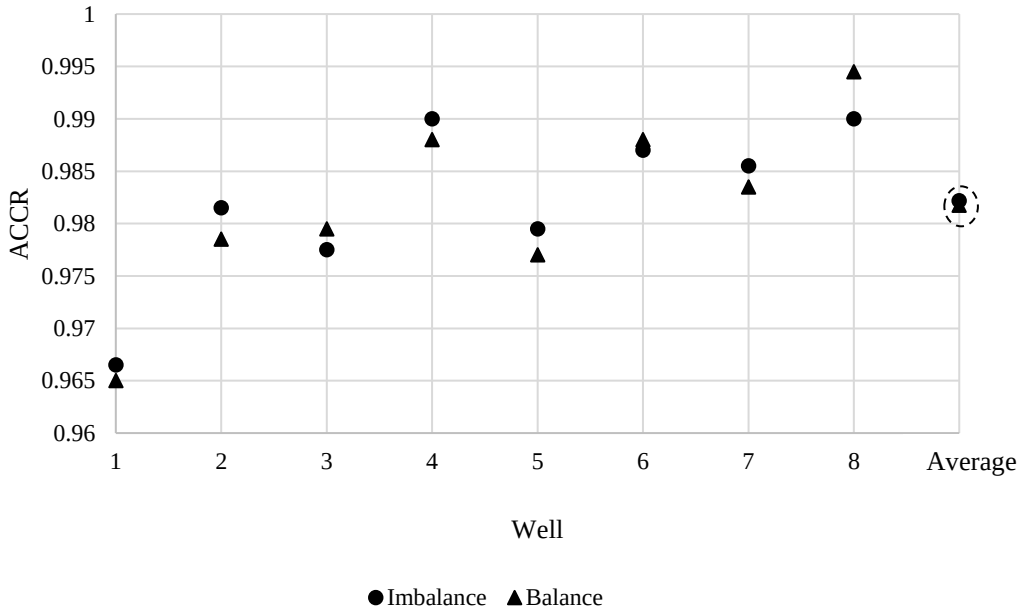


Figure 5.18. Comparison between average correct classification rate (ACCR) of discrimination of fractured/non-fractured zones by using balanced and imbalanced data over studied wells.

5.10 Ordered Weighted Averaging

In Tables 5.14 through 5.29 the details of classification by using two classifiers (RF and SVM) and imbalanced/balanced data are presented. Overall the results were fine; however, sometimes differences were considerable. For example, in Table 5.22, in which well 1 is test and balanced

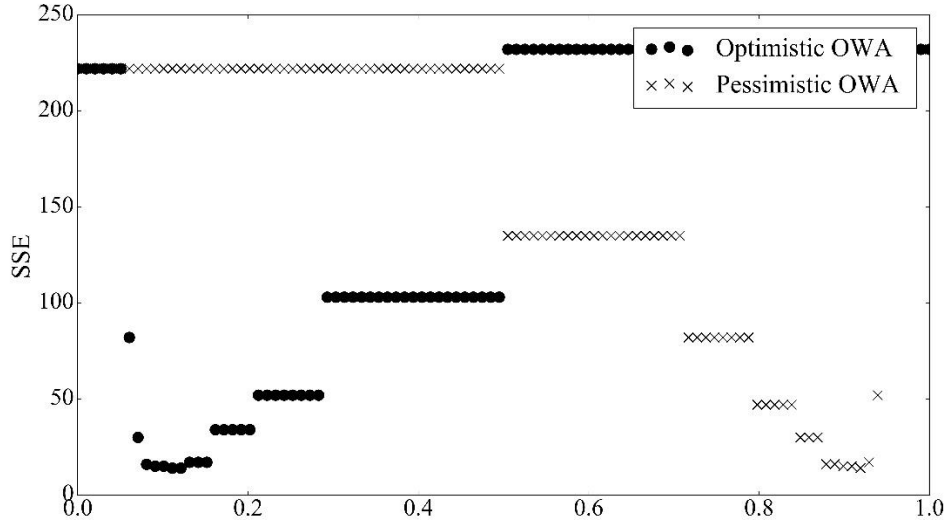
data were used for training of other wells, accuracy, precision and recall while well 3 was a training well were 98, 97 and 97 percent respectively (RF classifier). In this table, while well 6 was a training well, accuracy, precision and recall were 96, 94 and 97 percent respectively. In this case, both performance and harmony when well 3 was used as training are better.

Now, the question is when image logs are not available for a well, which well should be considered the training well? Ambiguity in decision-making leads to data fusion. Integrating the results achieved from different classifiers and all wells, to achieve more reliable, robust and accurate discrimination is the goal of data fusion. Based on the literature, OWA was selected for fusion. Optimistic (OOWA) and pessimistic (POWA) scenarios were applied over the 14 results of classifiers, and optimum α, which minimizes the sum of squared error (SSE), was found. It should be emphasized that SSE in the current study is equal to the number of misclassified data. In other words, SSE is equal to false negative (FN) plus false positive (FP) in Figure 5.4.

Results of optimization of α for all wells are presented in Figures 5.19 (for imbalanced data) and 5.20 (for balanced data). As can be seen in those figures, accuracy increased to about 100%. These are amazing results, in comparison to the best results (about 72%) that have been published so far [9, 10].

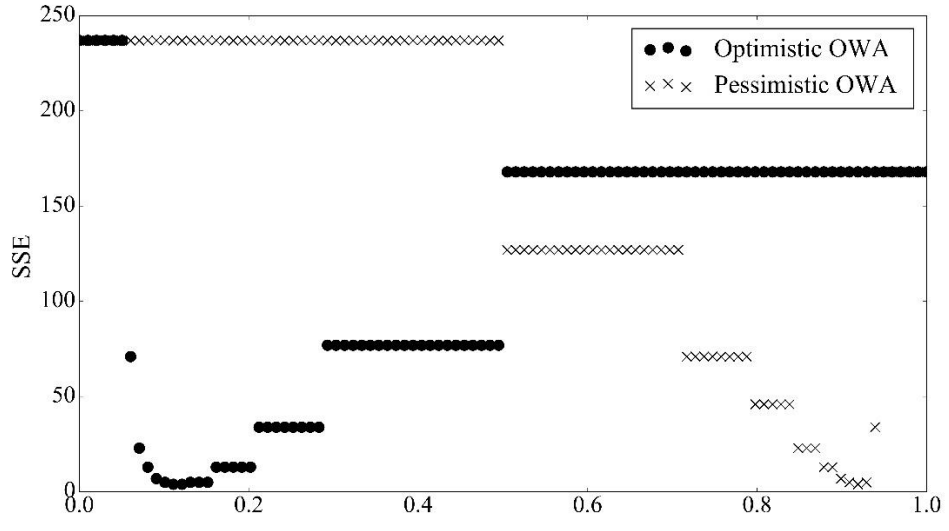
It should be emphasized that in all cases SSE for OOWA and POWA are equal (Figures 5.19 and 5.20); therefore, there is no priority for one of them. α_{opt} differs in the range of 0.09 to 0.13 in optimistic and 0.89 to 0.92 in pessimistic scenarios. In general, α_{opt} could be considered equal to 0.11 and 0.91 for optimistic and pessimistic scenarios respectively.

Code that is written for OWA is in appendix M.



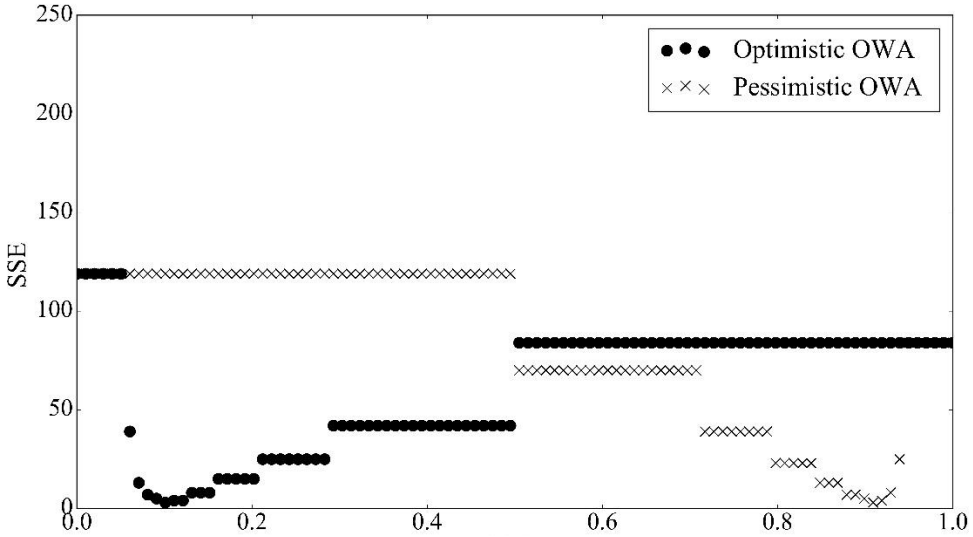
Well # 1

OOWA			POWA		
$\alpha_{opt} = 0.12$	No. of Errors = 14	$CCR = \frac{(2159-14)}{2159} \approx 1$	$\alpha_{opt} = 0.91$	No. of Errors = 14	$CCR = \frac{(2159-14)}{2159} \approx 1$



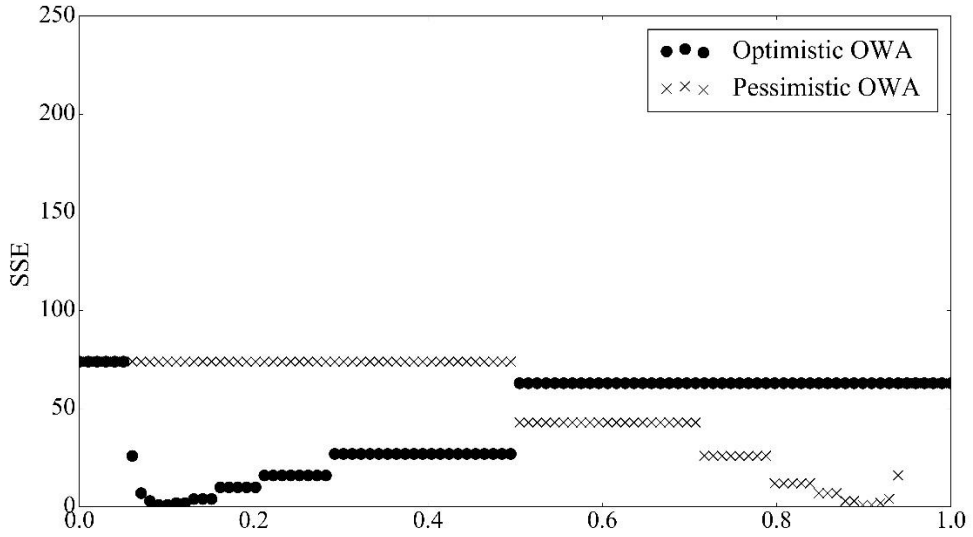
Well # 2

OOWA			POWA		
$\alpha_{opt} = 0.12$	No. of Errors = 4	$CCR = \frac{(3057-4)}{3057} \approx 1$	$\alpha_{opt} = 0.91$	No. of Errors = 4	$CCR = \frac{(3057-4)}{3057} \approx 1$



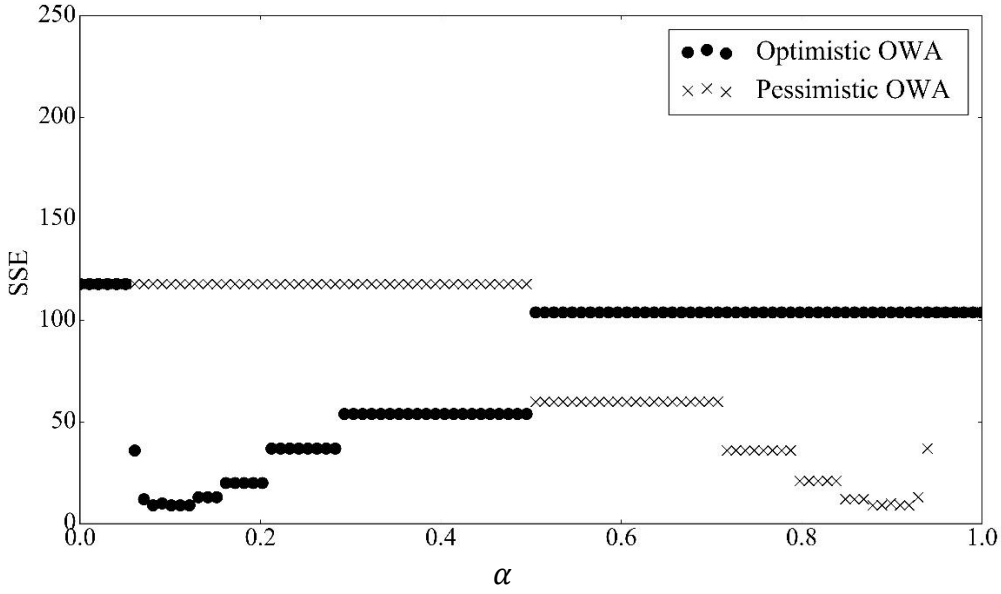
Well # 3

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 3	$CCR = \frac{(1449-3)}{1449} \approx 1$	$\alpha_{opt} = 0.90$	No. of Errors = 3	$CCR = \frac{(1449-3)}{1449} \approx 1$



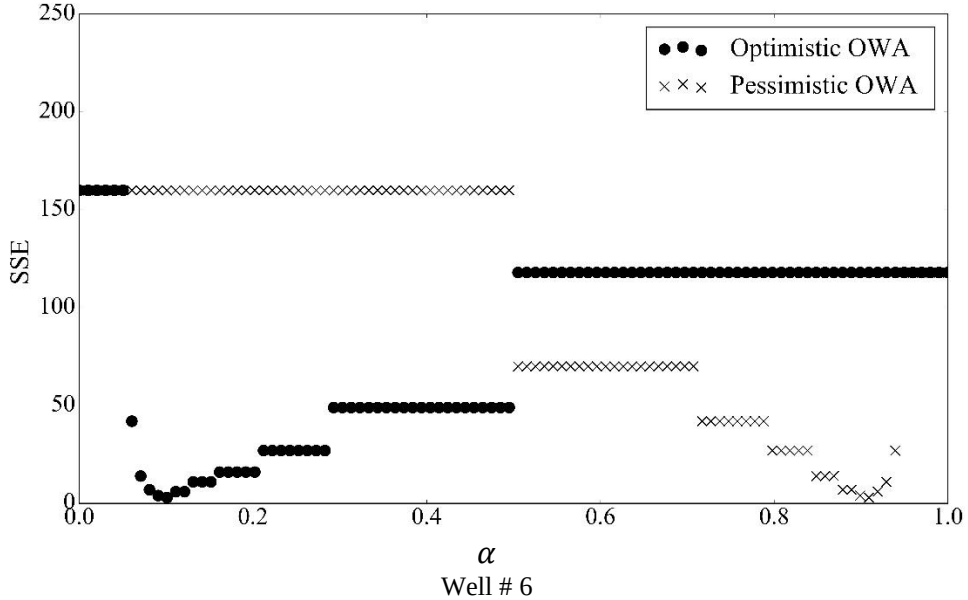
Well # 4

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 1	$CCR = \frac{(2041-1)}{2041} \approx 1$	$\alpha_{opt} = 0.90$	No. of Errors = 1	$CCR = \frac{(2041-1)}{2041} \approx 1$



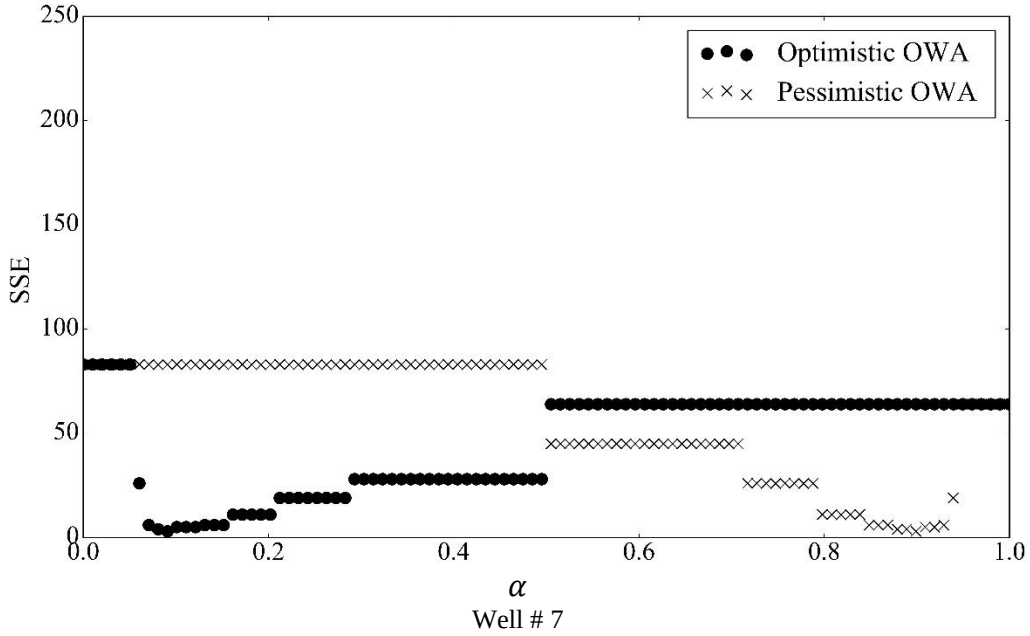
Well # 5

OOWA			POWA		
$\alpha_{opt} = 0.12$	No. of Errors = 9	$CCR = \frac{(1745-9)}{1745} \approx 1$	$\alpha_{opt} = 0.91$	No. of Errors = 9	$CCR = \frac{(1745-9)}{1745} \approx 1$



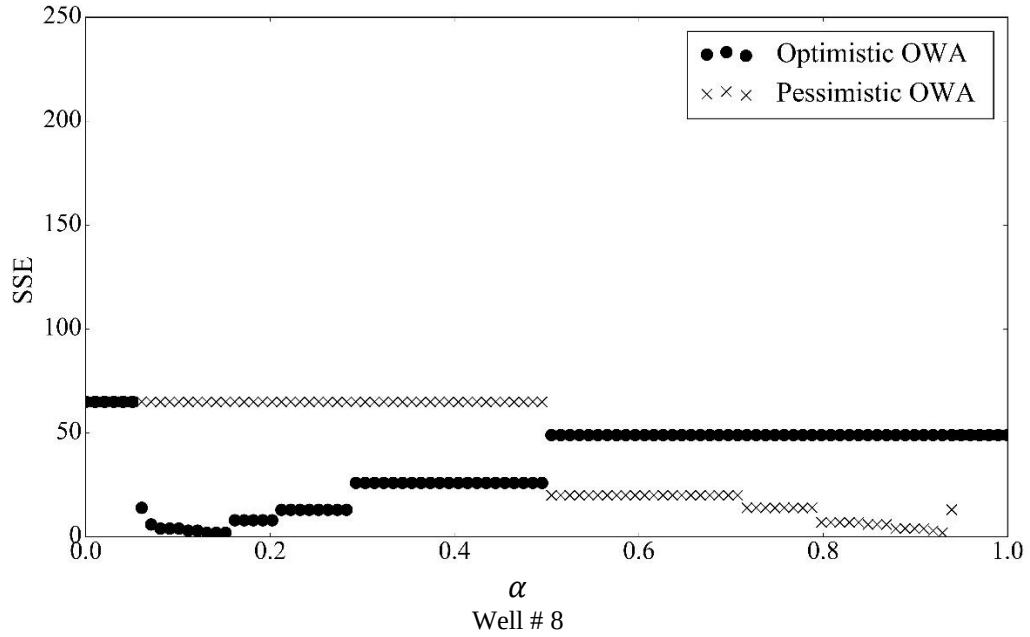
Well # 6

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 3	$CCR = \frac{(2845-3)}{2845} \approx 1$	$\alpha_{opt} = 0.90$	No. of Errors = 3	$CCR = \frac{(2845-3)}{2845} \approx 1$



Well # 7

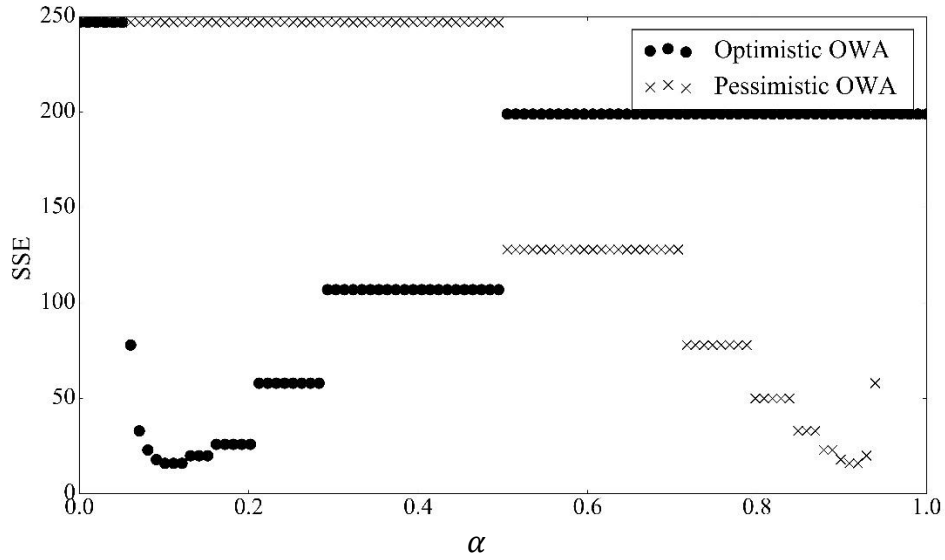
OOWA			POWA		
$\alpha_{opt} = 0.09$	No. of Errors = 3	$CCR = \frac{(1589-3)}{1589} \approx 1$	$\alpha_{opt} = 0.89$	No. of Errors = 3	$CCR = \frac{(1589-3)}{1589} \approx 1$



Well # 8

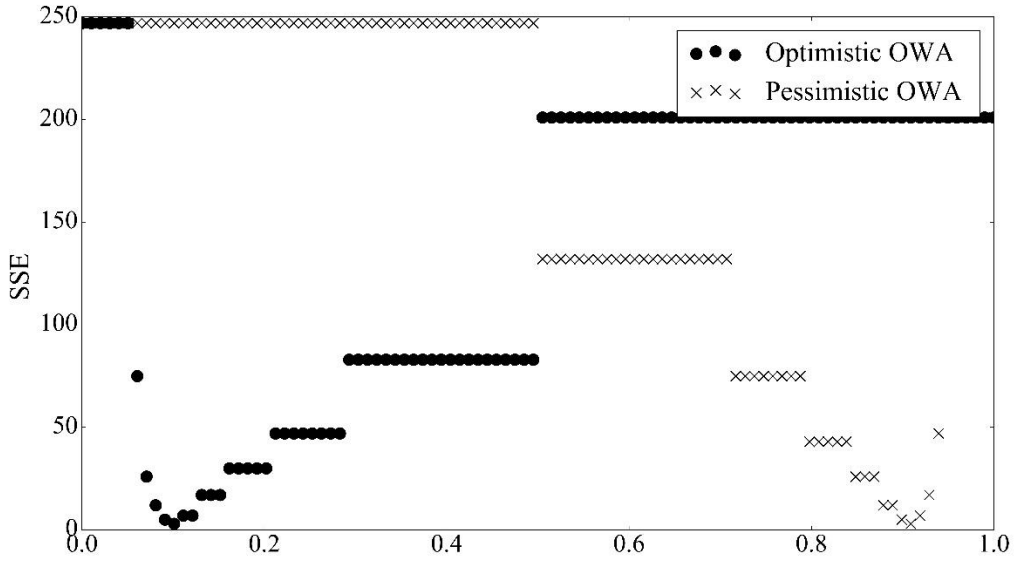
OOWA			POWA		
$\alpha_{opt} = 0.13$	No. of Errors = 2	$CCR = \frac{(2271-2)}{2271} \approx 1$	$\alpha_{opt} = 0.92$	No. of Errors = 2	$CCR = \frac{(2271-2)}{2271} \approx 1$

Figure 5.19 14 results of classifications by using IMBALANCED training data were fused using OWA data fusion method, in two scenarios (optimistic and pessimistic). The results of optimization of α, number of errors, and approximate CCR for all studied wells are presented.



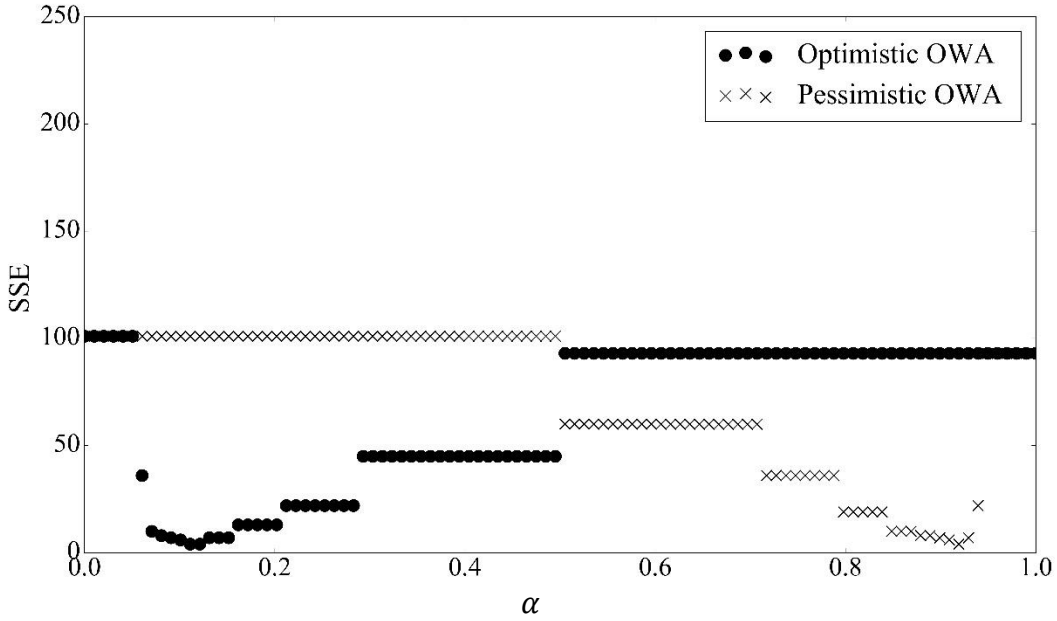
Well # 1

OOWA			POWA		
$\alpha_{opt} = 0.12$	No. of Errors = 16	$CCR = \frac{(2159-16)}{2159} \approx 1$	$\alpha_{opt} = 0.91$	No. of Errors = 16	$CCR = \frac{(2159-16)}{2159} \approx 1$



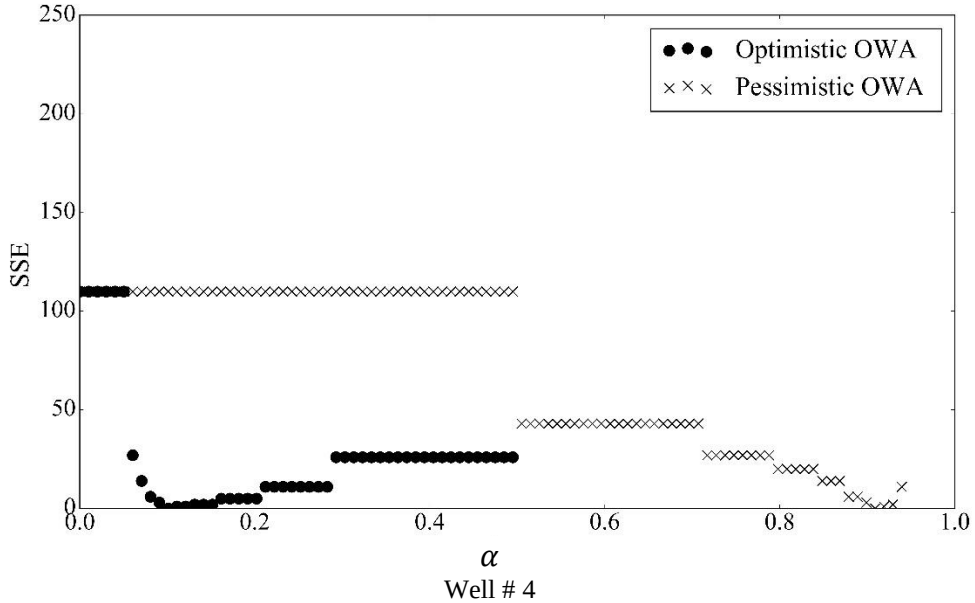
Well # 2

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 3	$CCR = \frac{(3057-3)}{3057} \approx 1$	$\alpha_{opt} = 0.90$	No. of Errors = 3	$CCR = \frac{(3057-3)}{3057} \approx 1$



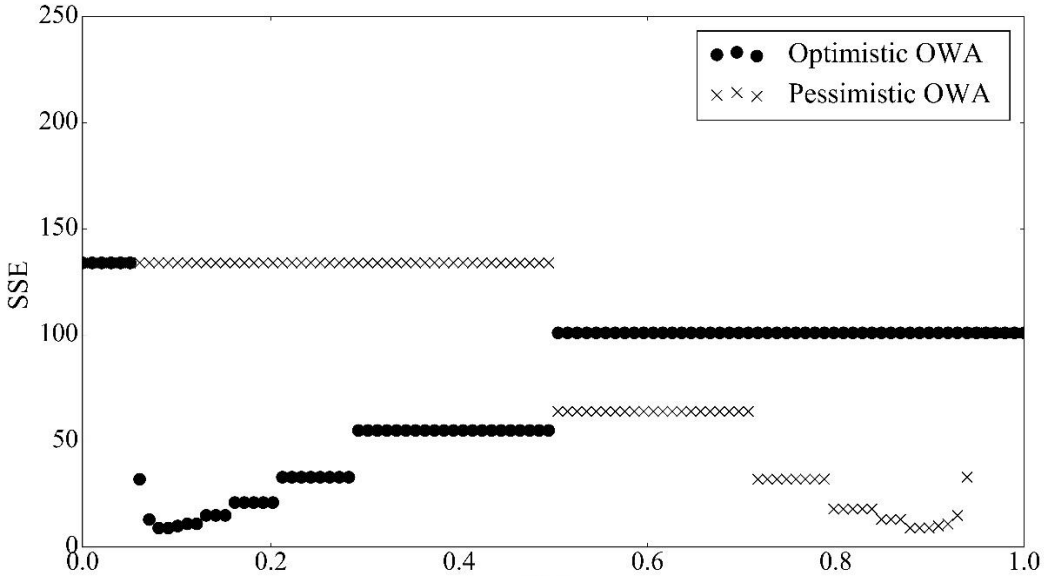
Well # 3

OOWA			POWA		
$\alpha_{opt} = 0.12$	No. of Errors = 4	$CCR = \frac{(1449-4)}{1449} \approx 1$	$\alpha_{opt} = 0.91$	No. of Errors = 4	$CCR = \frac{(1449-4)}{1449} \approx 1$



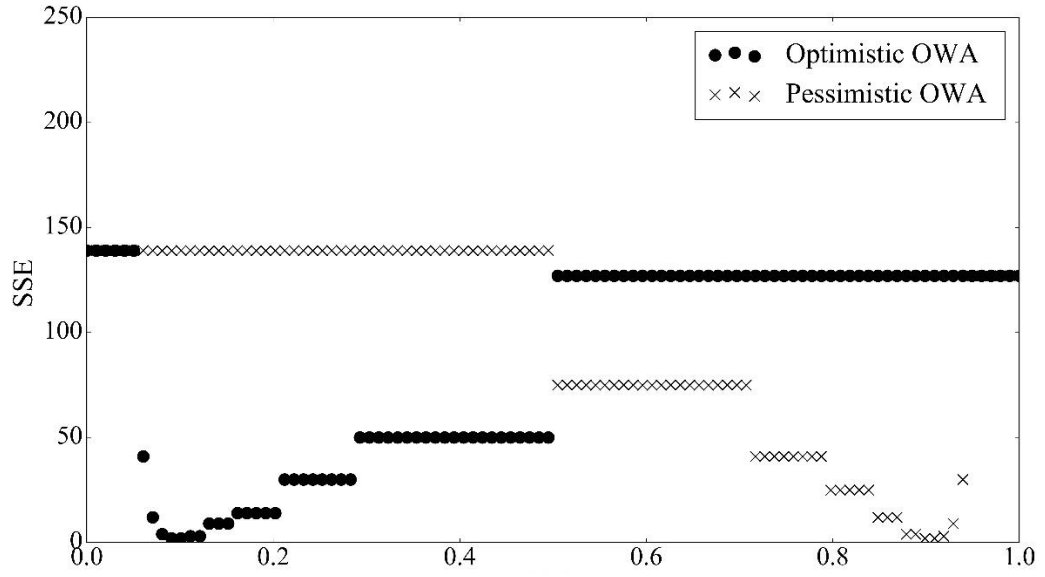
Well # 4

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 0	$CCR = \frac{(2041-0)}{2041} = 1$	$\alpha_{opt} = 0.90$	No. of Errors = 0	$CCR = \frac{(2041-0)}{2041} = 1$



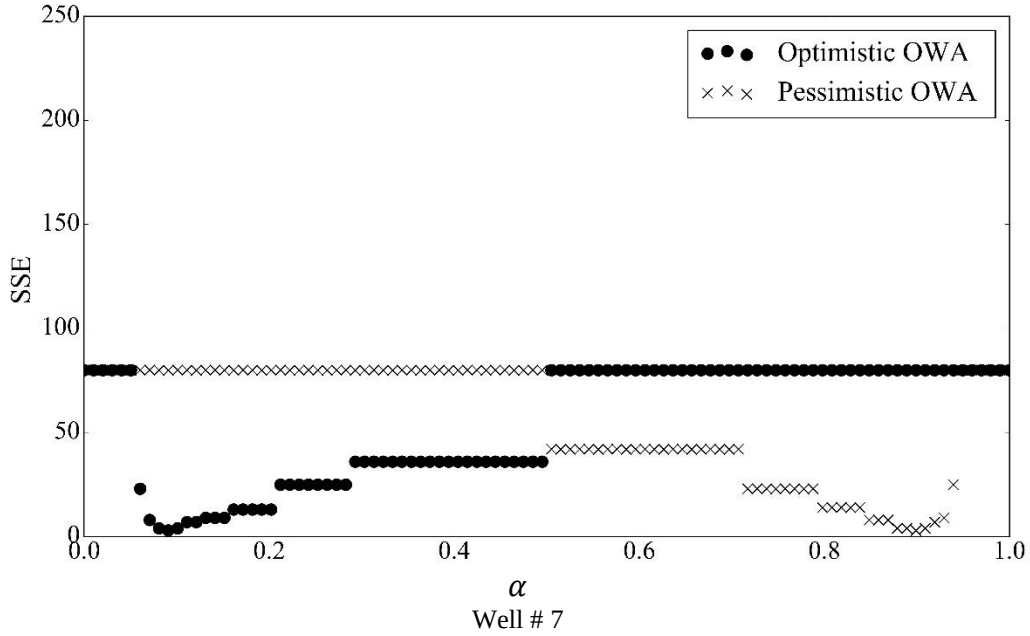
Well # 5

OOWA			POWA		
$\alpha_{opt} = 0.09$	No. of Errors = 9	$CCR = \frac{(1745-9)}{1745} \approx 1$	$\alpha_{opt} = 0.89$	No. of Errors = 9	$CCR = \frac{(1745-9)}{1745} \approx 1$



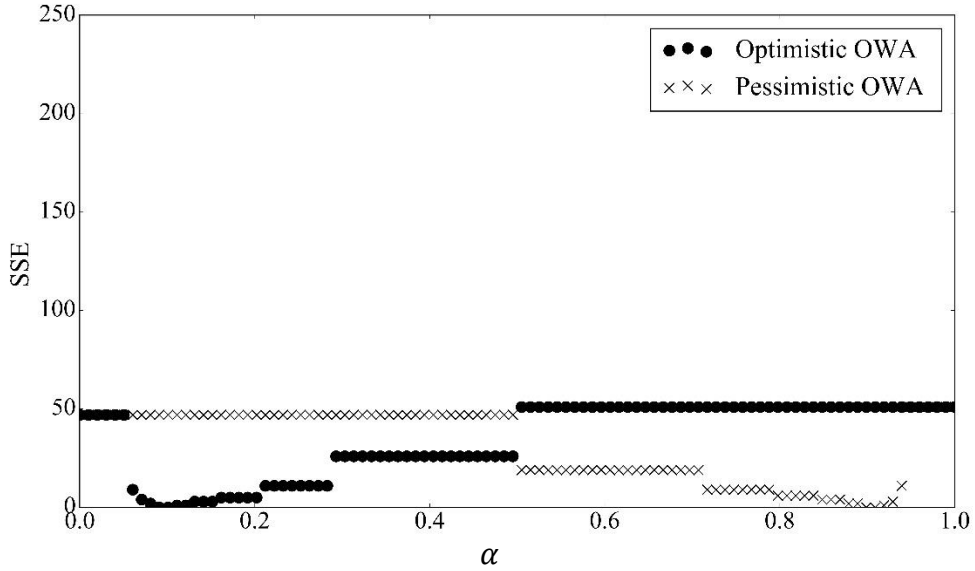
Well # 6

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 2	$CCR = \frac{(2845-2)}{2845} \approx 1$	$\alpha_{opt} = 0.90$	No. of Errors = 2	$CCR = \frac{(2845-2)}{2845} \approx 1$



Well # 7

OOWA			POWA		
$\alpha_{opt} = 0.09$	No. of Errors = 3	$CCR = \frac{(1589-3)}{1589} \approx 1$	$\alpha_{opt} = 0.89$	No. of Errors = 3	$CCR = \frac{(1589-3)}{1589} \approx 1$



Well # 8

OOWA			POWA		
$\alpha_{opt} = 0.10$	No. of Errors = 0	$CCR = \frac{(2271-0)}{2271} = 1$	$\alpha_{opt} = 0.90$	No. of Errors = 0	$CCR = \frac{(2271-0)}{2271} = 1$

Figure 5.20 14 results of classifications by using BALANCED training data were fused using OWA data fusion method, in two scenarios (optimistic and pessimistic). The results of optimization of α, number of errors, and approximate CCR for all studied wells are presented.

The average number of errors (FP plus FN) for RF, SVM, and OWA in balanced and imbalanced scenarios in Table 5.30 are reported. This table helps to rank the methods and kinds of the training database. Based on Table 5.30, accuracy of classifiers while using imbalanced data were a little bit better than the accuracy using balance data. Better results were achieved from balanced OWA. Table 5.30 also highlights that data fusion has improved the results more than 85 percent. Approximately, the summation of the number of errors for data fusion declined from about 300 to 40.

Table 5.30 Average FP plus FN for all utilized methods.

Well	Imbalance		Balance		Imbalance OWA	Balance OWA
	RF	SVM	RF	SVM		
1	78	72	75	72	14	16
2	51	63	61	73	4	3
3	36	33	30	31	3	4
4	14	27	25	30	1	0
5	35	38	32	43	9	9
6	37	42	38	38	3	2
7	24	23	24	29	3	3
8	18	14	15	11	2	0
Summation	293	312	300	327	39	37

5.11 ROC (Receiver Operating Characteristic) Curve Analysis

ROC (Receiver Operating Characteristic) curve is a fundamental graphical tool evaluation for a range of diagnostic test results. ROC curve plots the true positive rate (sensitivity) of a test versus its false positive rate (specificity) for different cut-off points of a parameter. ROC curve is a graphical plot to display discrimination of a binary classifier system by the trade-off relationship between sensitivity and specificity for all possible thresholds.

Sensitivity is the probability of a depth will be positive given as a fracture zone. Specificity is the probability of a depth will be negative given as a nonfracture zone. The accuracy of a test is measured by the area under the ROC curve (AUC). AUC is the area between the curve and the x-axis. An area of 1 represents a perfect test, while an area of .5 represents a worthless test. The closer the curve follows the left-upper corner of the plot, the more accurate the test.

The most appropriate threshold or cut-off for a classifier has the highest true positive rate together with the lowest false positive rate [86].

In figures 5.21-5.24, the ROC curve for both balance and imbalance data(one well as train and one well as a test), while SVM and RF classifiers are utilized, are shown respectively. (Appendix N & O)

As these figures show, ROC analysis provides important information about discrimination performance: the closer the apex of the curve toward the upper left corner, the greater the discriminatory ability of the classifier. This is measured quantitatively by the AUC such that a value of >0.96 indicates excellent discriminatory ability.

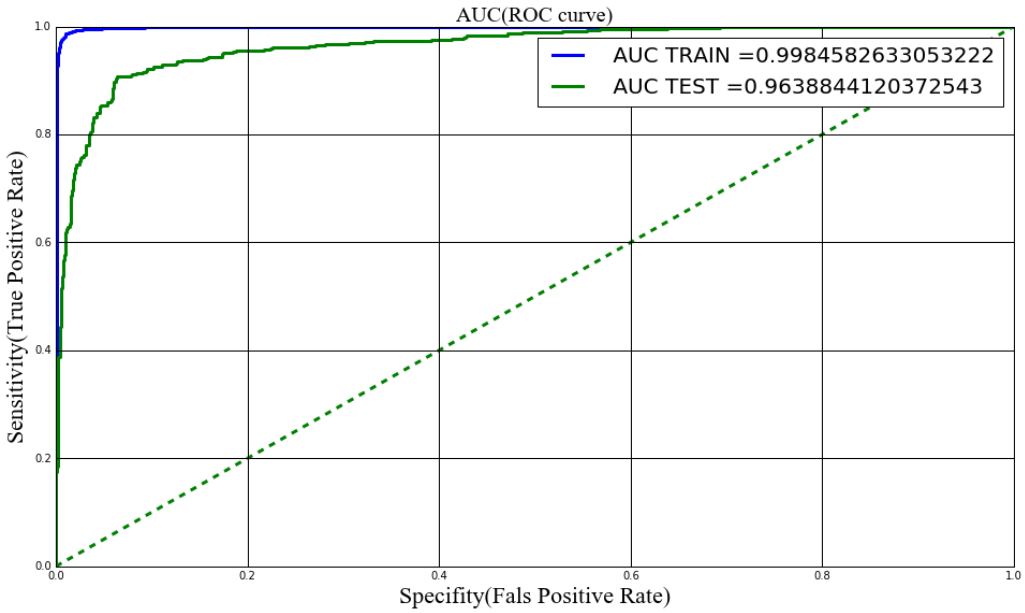


Figure 5.21 Receiver operating characteristic curve for SVM classifier on balanced data and while one well is as a train an another is as a test.

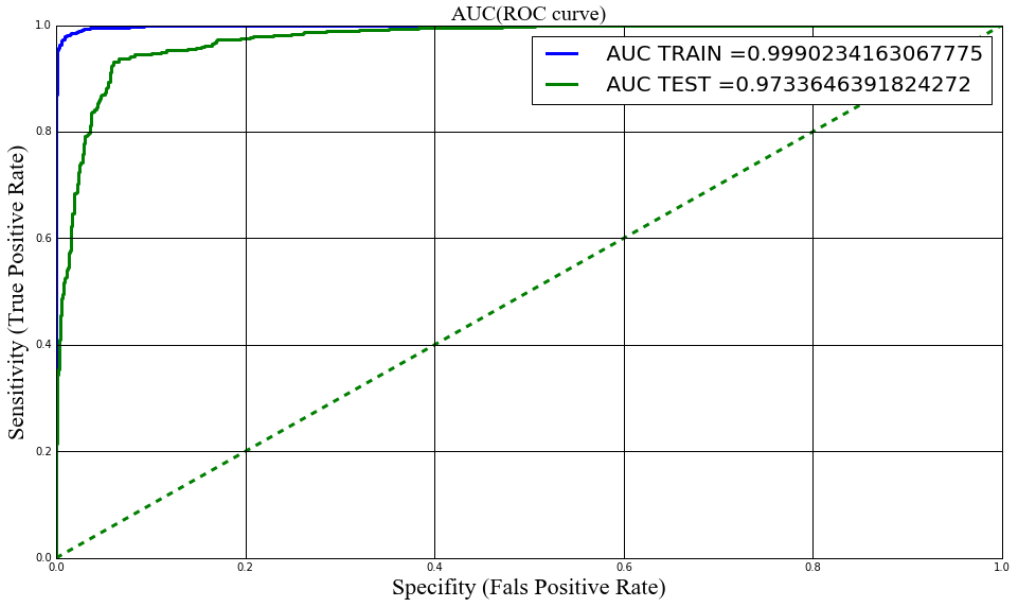


Figure 5.22 Receiver operating characteristic curve for SVM classifier on imbalanced data and while one well is as a train an another is as a test.

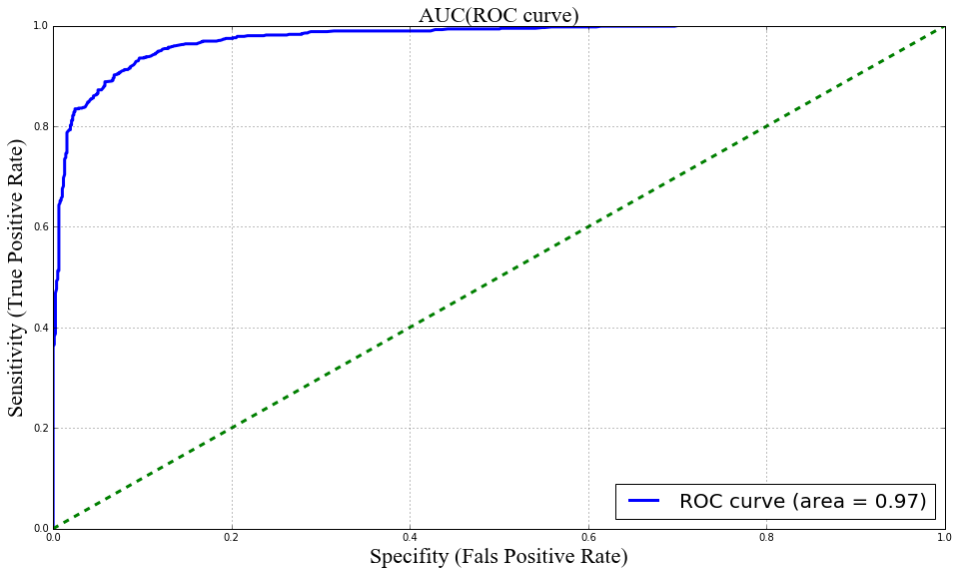


Figure 5.23 Receiver operating characteristic curve for RF classifier on balanced data and while one well is as a train an another is as a test.

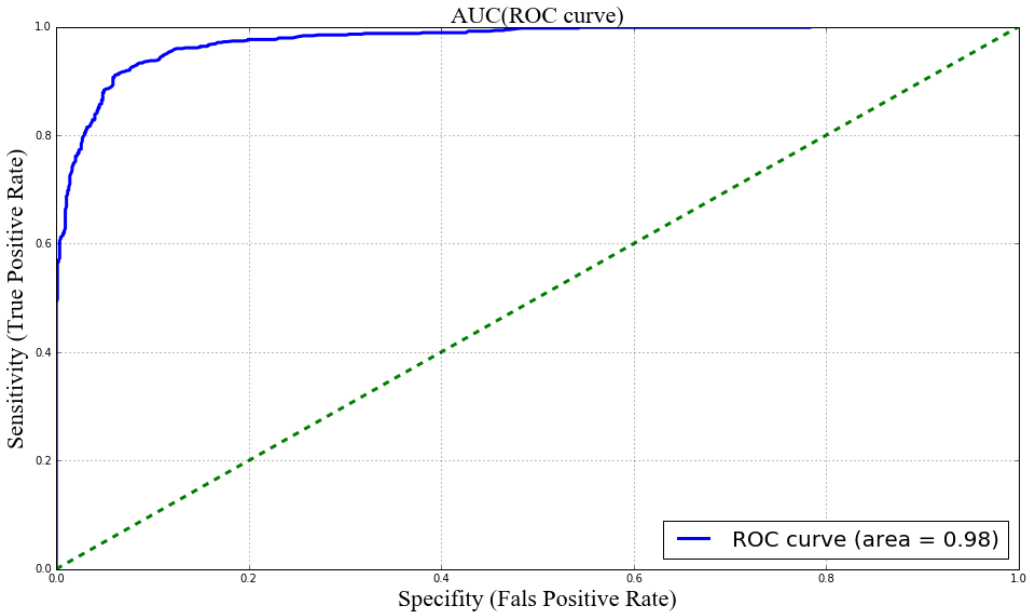


Figure 5.24 Receiver operating characteristic curve for RF classifier on imbalanced data and while one well is as a train an another is as a test.

6

Conclusions & Future Directions

6.1 Conclusions

In the current study image and petrophysical logs of a carbonate-fractured reservoir of a giant oil field in Iran were available. Therefore, conclusions dependent on the utilized database are not necessarily general. Contributions are briefly addressed as follows:

- Statistical studies showed that correlation coefficients between fractured zones and various logs differ in the range of -0.25 and 0.25. Hence, because of low correlation, fractured zone detection using one log is impossible.
- Statistical studies also showed that correlation coefficient between some of the logs are considerably high. In these cases, just one of the logs were selected. For instance, correlation coefficients between shale, CGR, potassium and thorium are often higher than 0.8, which is compatible with their physical behavior. CGR was selected in this case.

- Histograms of petrophysical logs in fractured/non-fractured zones showed that discrimination between those zones in 1D feature space (just using one of the petrophysical logs) is impossible. 2D and 3D graphs showed that better discrimination between fractured/non-fractured zones is possible by increasing the feature space. This leads to using classifiers for fractured zone detection in high feature space.
- 10 logs containing: caliper, CGR, SGR, RHOB, DT, PEF, NPHI, SW, effective and total porosity, were selected for fractured zone detection.
- Studies showed that in studied wells, the number of fractured zones are less than non-fractured ones. Imbalance index (UI) was defined to represent imbalanced situations in different wells. It showed that wells 6 (UI = 0.1) and 4 (UI = 5.73) are the most balanced and imbalanced wells respectively.
- Decision tree (DT), random forest (RF), support vector machine (SVM) and deep learning (DL) were applied to discriminate fractured zones from non-fractured ones by using selected petrophysical logs in each well. Database were split into train (70 %) and test (30 %) and classifications were utilized for imbalanced and balanced databases. Results showed that RF and SVM, with an average CCR of about 95%, give better discrimination in comparison with DT and DL. Also, imbalanced data give better accuracy in comparison to balanced ones.
- One of the most important outputs was about imbalanced wells (well 4: UI = 5.73 and well 3: UI = 2.18), in which the difference between average accuracy and precision using imbalanced data is high. This shows that classifiers have biases to non-fractured zones, and using balanced data is important. For example, in well 4, while using imbalanced

data, average accuracy was 15 percent higher than average precision, which dropped to about 2 percent by switching to balanced data.

- RF and SVM were selected for the generalization step, in which classifiers were tested in one of the wells and trained by the other wells. In this step, average accuracy was about 98 percent. These are unbelievable results. CCRs for imbalanced datasets were a little bit better than balanced ones.
- In the generalization stage, there was not an observed considerable difference between the results of balanced and imbalanced data. Therefore, it was concluded that classification during generalization is not sensitive to balancing of the training data.
- Examining the results showed that errors occurred in the boundaries of fractured/non-fractured zones. In the industry, fractured zone recognition is important, not the resolution of boundaries. So, achieved results are highly encouraging.
- In the generalization step, 14 results were achieved for each well. Seven results were obtained when RF was utilized for training of 7 wells, and the classifier was applied to the remaining well, which was used as test well. Seven similar results were achieved while using SVM. Ordered weighted averaging was applied to integrate 14 results together, in order to achieve one unique, more accurate and reliable result. The accuracy for outputs for all wells were about 100 percent, amazing! It should be emphasized that the best accuracy in previous studies was about 72 %.
- In all cases SSE for optimistic and pessimistic OWA were similar. 0.11 and 0.91 were reported as α_{opt} for optimistic and pessimistic scenarios respectively.

- On average, the summation of the number of errors for data fusion declined from about 300 to 40. So, data fusion decreases the number of errors of classifiers by more than 85 percent.

6.2 Possible Future Directions

Fractures contain more than 10 properties, the recognition of some of which are highly important in oil reservoirs. For example, fracture could be open or closed. Open fractures are a high permeable path for fluids, while closed ones are a barrier. Aperture, roughness, spacing, length, filling and filler of open fractures have a considerable effect on porosity, permeability and stability of the reservoir. Also, orientation (dip and dip direction) of fractures are the main factors in fracture network simulation.

In the current study, fractured zones were discriminated from non-fractured ones. The accuracy of discrimination was about 100 percent. Now the question is how open fractured zones could be discriminated from closed fractured zones. Or how an estimation of aperture, or other properties would be assigned to open fractures. Answers to those kinds of questions will help in making semi-image logs, which are so beneficial, and might be the topic of future research.

References

[1] US Energy Information Administration

[2] http://www.bris.ac.uk/research/impact/carbonate-reservoirs/#:~:text=Extracting%20oil%20fram%20carbonate%20reservois,types%20of%2035%20per%20cent

[3] Nelson R.A. (2001) Geologic Analysis of Naturally Fractured Reservoirs, SECOND EDITION, Gulf Professional Publishing, U.S.A., 332 Pages.

[4] Martinez-Torres L.P. (2002) Characterization of Naturally Fractured Reservoirs from Conventional Well Logs, M.Sc. Thesis, University of Oklahoma. USA.

[5] Dutta P. Singh S.K. Al-Genai J. Akhtar A. and Akbar M. (2007) A Novel Approach to Fracture Characterization Utilizing Borehole Seismic Data, 15th SPE Middle East Oil & Gas Show and Conference held in Bahrain International Exhibition Centre, Kingdom of Bahrain, 11-14 March 2007, Paper SPE 105427.

[6] Schlumberger (1998) Log Interpretation Principles/Applications, Seven Printing, March Sugar Land, Texas 77478.

[7] Thompson L.B. (2000) Fractured Reservoirs: Integration is the Key of Optimization, Distinguished Author Series, 52-54, SPE 56010.

[8] Serra O. (1989) Formation MicroScanner Image Interpretation, Schlumberger Education Services.

[9] Tokhmechi B. Memarian H. Ahmadi Noubari H. Moshiri B. (2009) A novel approach for fracture Zone detection using petrophysical logs, Journal of Geophysics and Engineering, **6**, 365-373.

[10] Mazhari S.M. Memarian H. *Tokhmechi B.* (2018) A Hybrid Learning Automata and Case-Based Reasoning for Fractured Zone Detection, Arabian Journal of Geosciences, Doi.org/10.1007/s12517-018-3934-3.

[11] Roehl P.O. and Choduette P.W. (1985) Carbonate Petroleum Reservoirs, New York, Springer-Verlag, 622 Pages.

[12] Abyat Y. Abyat A. Abyat A. (2019) Microfacies and depositional environment of Asmari formation in the Zeloi oil field, Zagros basin, south-west Iran, Carbonates and Evaporites **34**, 1583–1593.

[13] Amirshahkarami M. Vaziri-Moghaddam H. Taheri A. (2007) Sedimentary facies and sequence stratigraphy of the Asmari Formation at Chaman-Bolbol, Zagros Basin, Iran, Journal of Asian Earth Sciences, **29**(5-6) 947-959.

[14] Alavi M. (2004) Regional stratigraphy of the Zagross fold-thrust belt of Iran and its proforeland evolution, American Journal OF Science, **304**, 1-20.

[15] Farahzadi E. Alavi S.A. Sherkati S. Ghasemi M.R. (2019) Variation of subsidence in the Dezful Embayment, SW Iran: influence of reactivated basement structures, Arabian Journal of Geosciences, 12. 616. DOI: https://doi.org/10.1007/s12517-019-4758-5.

[16] McQuillan H. (1985) Fracture-Controlled Production from the Oligo-Miocene Asmari Formation in Gachsaran and Bibi Hakimeh Fields, Southwest Iran. In: Roehl P.O., Choquette P.W. (eds) Carbonate Petroleum Reservoirs. Casebooks in Earth Sciences. Springer, New York, NY.

[17] Vatandoust M. Saein A.F. (2017) Production of open fractures in the Asmari formation using geometrical analysis: Aghajari anticline, Dezful Embayment, SW IRAN, **40**(4) 413-426.

[18] Chow E. Ashayeri C. Stanely A.J. (2018) The Future of Iran's Oil and Gas Industry, USAEE Working Paper No. 18-364, 26 Pages.

[19] Murray G.H. (1968) Quantitative fracture study – Spanish pool, McKenzie County, North Dakota: Am. Assoc. Petroleum Geologists Bull., **52**, 57-65.

[20] Nemcok M. Lisle, R.J. (1995) A stress inversion procedure for polyphase fault/slip data sets, Journal of Structural Geology, **17**(10), 1445-1453.

[21] McQuillan H. (1991) The role of basement tectonics in the control of sedimentary facies, structural patterns and salt plug emplacements in the Zagros Fold belt of SW Iran. Journal of Southeast Asian Earth Sciences, **5**, 453-463.

[22] Cosgrove J.W. (2015) The association of folds and fractures and the link between folding, fracturing and fluid flow during the evolution of a fold–thrust belt: a brief review, Geological Society, London, Special Publications, **421**, 41-68.

[23] McQuillan H. (1973) Small-scale fracture density in Asmary Formation of southwest Iran and its relation to bed thickness and structural setting, The American Association of Petroleum Geologists Bulletin (AAPG), **57**(12), 2367-2385.

[24] Colman-Sadd S.P. (1978) Fold development in Zagros simply folded belt, SW Iran. American Association of Petroleum Geologists Bulletin, **62**, 984-1003.

[25] Rezaie A. H. and Nogole-Sadat M.A. (2004) Fracture Modeling in Asmari Reservoir of Rag-e Sefid Oil-Field by using Multiwell Image Log (FMS/FMI), Iranian Int. J. Sci. **5**(1), 107-121

[26] Charles H. Blumentritt Kurt J. Marfurt and Charlotte Sullivan E. (2006) Volume Based Curvature Computations Illuminate Fracture Orientations, Lower-Mid Paleozoic, Central Basin Platform, West Texas, Allied Geophysical Laboratories, University of Houston.

[27] Motiei H. (1996) Geology of Iran, Oil Geology of Zagros, Geological Survey & Mineral Exploration of Iran, 589 pages.

[28] Watson S.E. (1960) Studying wells in Gachsaran Oil Field, Iranian Oil Consortium, Private Report.

[29] Henson F.R.S. (1951) Observations on the geology & petroleum occurrences of the Middle East, Proceeding of the 3rd world petroleum congress, **1**, 118-140.

[30] McCord D.R. and Associates (1970-75) Geological evaluation and computer reservoir prediction of future performance of oil fields in North Africa (Algeria), Australia (Cooper Creek Basin), Middle East (Kuwait, Abu Dubai, Saudi Arabia, Iran) and various North American oil fields, Dallas, Texas.

[31] Sepehr M. and Cosgrove J.W. (2004) Structural framework of the Zagros Fold–Thrust Belt, Iran, Marine and Petroleum Geology **21**, 829–843.

[32] Nemati M. Pezeshk H. (2005) Spatial Distribution of Fractures in the Asmari Formation of Iran in Subsurface Environment: Effect of Lithology and Petrophysical Properties, Natural Resource Research, **14**(4), 305-316

[33] Falcon N.L. (1969) Problems of the relationship between surface structure and deep displacements illustrated by the Zagros range, Geological Society of London, 9-22.

[34] Edgell H.S. (1992) Basement tectonic of Saudi Arabia as related to field structures: In Rickard et al. (eds.), Basement Tectonic, Kluwer Academic Publishers, Dordrecht, 169-193.

[35] Ameen M.S. (1992) Effect of basement tectonics on hydrocarbon generation, migration and accumulation in North Iraq, AAPG Bulletin, **76**, 356-370.

[36] Dashti R. Bagheri M.B. Ulhaq S. (2009) Fracture Characterizing and Modeling of a Porous Fractured Carbonate Reservoir, SPE 125329, SPE/EAGE Reservoir Characterization and Simulation Conference, At Abu Dhab, 17 pages.

[37] Sangree J.B. Masson P.H. McQuillan H. and Twerenbold E.F. (1961) Preliminary notes on Asmari fracture study, Iranian Oil Consortium, Private Report.

[38] McQuillan H. (1974) Fracture Patterns on Kuh-e Asmari Anticline, Southwest Iran, The American Association of Petroleum Geologists Bulletin (AAPG), **58**(2), 236-245.

[39] Gholipur A.M. (1994) Patterns and structural positions of productive fractures in the Asmari Reservoirs, Southwest Iran. Proceedings of the Canadian SPE/ CIM/ CANMET Conference, Recent Advances in Horizontal Well Applications, Calgary, Paper HWC94-43, 10.

[40] Wennberg O.P. Svana T. Azizzadeh M. Aqrawi A.M.M. Brockbank P. Lyslo K.B. and Ogilvie S. (2006) Fracture intensity vs. mechanical stratigraphy in platform top carbonates: the Aquitanian of the Asmari Formation, Khaviz Anticline, Zagros, SW Iran Petroleum Geoscience, August 1, **12**(3), 235-246.

[41] Wennberg O.P. Azizzadeh M. Aqrawi A.A.M. Blanc E. Brockbank P. Lyslo K.B. Pickard N. Salem L.D. and Svana T. (2007) The Khaviz Anticline: an outcrop analogue to giant fractured Asmari Formation reservoirs in SW Iran, Geological Society, London, Special Publications, **270**(1), 23–42.

[42] Stephenson B.J. Koopman A. Hillgartner H. McQuillan H. Bourne S. Noad J.J. and Rawnsley K. (2007) Structural and stratigraphic controls on fold-related fracturing in the Zagros Mountains, Iran:

implications for reservoir development Geological Society, London, Special Publications, January 1, **270**(1), 1-21.

[43] Ahmadhadi F. Lacombe O. and Daniel J.M. (2007) Early reactivation of basement faults in Central Zagros (SW Iran): Evidence from pre-folding fracture patterns in Asmari Formation and lower Tertiary Paleogeography, in trust belts and foreland basins, Springer, **11**, 205-228.

[44] Sahimi M. and Hashemi M. (2001) Wavelet Identification of the Spatial Distribution of Fractures, Geophysical Reservoir Letters, **28**(4), 611-614.

[45] Tokhmechi B. (2009) Identification and categorization of joints using data fusion approach, emphasizing on Asmari Formation, PhD Thesis, University of Tehran, 165 pages.

[46] Nemati M. and Pezeshk H. (2005) Spatial distribution of fractures in the Asmari formation of Iran in subsurvace environment: effect of lithology and petrophysical properties. Natural Resources Research, **14**, 305-316.

[47] Mohebbi A.R. Haghighi M. Sahimi M. (2007) Using Conventional Logs for Fracture Detection and Characterization in One of Iranian Field. International Petroleum Technology Conference held in Dubai, U.A.E., 4-6 December 2007, Paper IPTC 11186.

[48] Hsu K. Brie A. and Plumb R.A. (1987) A New Method for Fracture Identification Using Array Sonic Tools, Journal of Petroleum Technology, SPE 14397, 677-683.

[49] Flavio S.A. and Gregor P.E. (1999) The Velocity-deviation log a tool to predict pore type and permeability trends in carbonate drill holes from sonic and porosity or density logs, The American Association of Petroleum Geologists Bulletin, **83**(3), 450–466.

[50] Daiguji M. Kudo O. and Wada T. (1997) Application of Wavelet Analysis to Fault Detection in Oil Refinery, Computers Chem Engng, **21**, S 1117-S 1122.

[51] Behrens R.A. Macleod M.K. Tran T.T. and Alimi A.O. (1998) Incorporating Seismic Attribute Maps in 3D Reservoir Models, SPE Reservoir Evaluation, **1**, 122-126.

[52] Song X. zhu Y. Iiu Q. Chen J. Ren D. Li Y. Wang B. and Liao M. (1998) Identification and Distribution of Natural Fractures, SPE International Conference and Exhibition in China held In Beijing, Paper SPE 50877.

[53] Tran N.H. (2004) Characterization and Modeling of Naturally Fractured Reservoirs, Ph.D. Thesis, University of New South Wales, Australia.

[54] Ozkaya S.I. and Siyabi S. (2008) Detection of Fracture Corridors from Dynamic Data by Factor Analysis, SPE Saudi Arabia Section Technical Symposium held in Alkhobar, Paper SPE 120812.

[55] Yan J. Lu L. Lubbe R. and Payne S. (2009) Petrophysical fracture identification for rock physics studies, 71st EAGE Conference & Exhibition (Amesterdam), 5 pages.

[56] Tokhmechi B. Memarian H. Rasouli V. Ahmadi Noubari H. Moshiri B. (2009) Fracture Zones Detection Using Wavelet Decomposition of Water Saturation Log, Journal of Petroleum Science and Engineering, **69**, 129-138.

[57] Tokhmechi B. Memarian H. Rezaee M.R. (2010) Estimation of the fracture density in fractured zones using petrophysical logs, Journal of Petroleum Science and Engineering, **72**, 206-213.

[58] Jafari A. Kadkhodaei A. Sharghi Y. and Ghanavati K. (2012) Fracture density estimation from petrophysical log data using the adaptive neuro-fuzzy inference system, Journal of Geophysics and Engineering, 9(1), 105-114.

[59] Mazaheri A. Memarian H. Tokhmechi B. Najar Araabi B. (2015) Developing fracture measure as an index of fracture impact on well-logs, Energy Exploration & Exploitation, **33**(4), 555-574.

[60] Mazaheri A. Memarian H. Tokhmechi B. Najar Araabi B. (2019) Cell size optimization for fracture measure estimation in multi-scale studies within oil wells, Carbonates and Evaporites, DOI: 10.1007/s13146-017-0378-x, **34**(2), 261-272.

[61] Aghli G. Soleimani B. Moussavi-Harami R. and Mohammadian R. (2016) Fractured zones detection using conventional petrophysical logs by differentiation method and its correlation with image logs, Journal of Petroleum Science and Engineering, 142, 152-162.

[62] Zarehparvar Ghoochaninejad H., Asef M.R. and Moallemi S.A. (2018) Estimation of fracture aperture from petrophysical logs using teaching–learning-based optimization algorithm into a fuzzy inference system, Journal of Exploration and Production Technology, **8**, 143-154.

[63] Moradi M., Tokhmechi B. and Masoudi P. (2019) Inversion of well logs into rock types, lithofacies and environmental facies, using pattern recognition, a case study of carbonate Sarvak Formation, Carbonates and Evaporites, DOI: 10.1007/s13146-017-0388-8, **34**(2), 335-347.

[64] Olivia Dias L. Bom C. R. Faria E.L. Valentín M.B. Correia M.D. Albuquerque M.P. de Albuquerque M.P. Coelho J.M. (2020) Automatic detection of fractures and breakouts patterns in acoustic borehole image logs using fast-region convolutional neural networks, Journal of Petroleum Science and Engineering, 191, 10 pages.

[65] Schlumberger Log Interpretation Principles/Applications. (1998) Seven Printing. March Sugar Land, Texas 77478.

[66] McQuillan, H. (1985) Fracture-controlled production from the Oligo-Miocene Asmari Formation in Gachsaran and Bibi Hakimeh Fields, SW Iran. In: ROEHL, P. O. & CHOQUETTE, P. W. (eds) Carbonate Petroleum Reservoirs. Springer, Berlin, 513-523.

[67] Halliburton (1995) Electrical Micro Imaging Service, Sales Kit, SK 1011.

[68] Schlumberger (2002) Borehole geology, geomechanics and 3D reservoir modeling (FMI), SMP-5822.

[69] Schlumberger (1993) Azimuthal Resistivity Imager (ARI), SMP-9260.

[70] Schlumberger (2001) Borehole Imaging in Oil-Base Mud (OBMI), SMP-5806.

[71] Schlumberger (2002) Advanced Borehole imaging independent of mud type (UBI), SMP-5871.

[72] Fathi N. Delshad M. Sepehrnoori K. Nguyen Q.P. and Zhang J. (2008) Chemical Flooding of Fractured Carbonates Using Wettability Modifiers, SPE/DOE Improved Oil Recovery Symposium held in Tulsa, Oklahoma, U.S.A., 19 Pages, Paper SPE 113369.

[73] Overview About The Decision Tree Model: https://medium.com/analytics-idhya/ overview-about-the-decision-tree-model-267c870fa147

[74] Decision Tree Classification in Python: https://www.datacamp.com/community/tutorials/ decision-tree-classification-python

[75] A Beginner's Guide to Important Topics in AI, Machine Learning, and Deep Learning. https://wiki.pathmind.com/multilayer-perceptron

[76] Dive into Deep Learning, https://d2l.ai/index.html

[77] Brain structure and development, https://activekidsphysiotherapy.com.au/brain-structure-and-development/

[78] Computer Vision: A Study on Different CNN Architectures and their Applications, https://medium.com/ alumnaiacademy/introduction-to-computer-vision-4fc2a2ba9dc

[79] ML - Support Vector Machine (SVM), https://www.tutorialspoint.com/machinelearning withpython/machinelearningwithpythonclassificationalgorithmssupportvectormachine.htm

[80] SVM classifier, Introduction to support vector machine algorithm, https://dataaspirant.com/support-vector-machine-algorithm/#comments

[81] Yager, R.R. (1988) On ordered weighted averaging aggregation operators in multi criteria decision making, IEEE Transactions on Systems, Man, and Cybernetics, **18**(1) 183-190.

[82] Hsu, H.M., Chen, C.T. (1996) Aggregation of fuzzy opinions under group decision making environment, Fuzzy Sets and Systems, **79**, 279-285.

[83] Kuncheva, L.I., Krishnapuram, R. (1996) A fuzzy consensus aggregation operator, Fuzzy Sets and Systems, **79**, 347-356.

[84] Theodoridis S. and Koutroumbos K. (2002) Pattern Classification, 2nd edition (San Diego: Elsevier/Academic) 710 pages.

[85] Wei-Meng L. (2019) Python- Machine Learning, Getting Started with Scikit-learn for Machine Learning, Chapter 5, John Wiley & Sons, Inc. 93-117. DOI: 10.1002/9781119557500.ch5.

[86] Hand D.J. and Till R.J. (2001). "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems." *Machine Learning* 45(2), p. 171-186.

Appendix A

Code: Decision Tree

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
zeros = df119.loc[(df119['Fracture'] == 0)].sample(n = 945)
ones = df119.loc[(df119['Fracture'] == 1)]
from IPython.display import display
from IPython.display import Image
dfbalance = pd.concat([zeros, ones])
dfbalance
from sklearn.model_selection import train_test_split
X1 = dfbalance.drop('CALI', axis=1)
X = X1.drop('Fracture', axis=1)
y = X1['Fracture']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree.fit(X_train,y_train)
predictions = dtree.predict(X_test)
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=500)
rfc.fit(X_train, y_train)
rfc_pred = rfc.predict(X_test)
print(confusion_matrix(y_test,rfc_pred ))
print('\n')
print(classification_report(y_test,rfc_pred ))
```

Appendix B

Code: Random Forest

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
zeros = df119.loc[(df119['Fracture'] == 0)].sample(n = 945)
ones = df119.loc[(df119['Fracture'] == 1)]
from IPython.display import display
from IPython.display import Image
dfbalance = pd.concat([zeros, ones])
dfbalance
from sklearn.model_selection import train_test_split
X1 = dfbalance.drop('CALI', axis=1)
X = X1.drop('Fracture', axis=1)
y = X1['Fracture']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=500)
rfc.fit(X_train, y_train)
rfc_pred = rfc.predict(X_test)
print(confusion_matrix(y_test,rfc_pred ))
print('\n')
print(classification_report(y_test,rfc_pred ))
```

Appendix C

Code: Decision Tree Visualization

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
# In[2]:
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
# In[3]:from sklearn.tree import export_graphviz
# In[4]:from sklearn.model_selection import train_test_split
# In[5]:X1 = df119.drop('CALI', axis=1)
# In[6]:X = X1.drop('Fracture', axis=1)
# In[7]:y = X1['Fracture']
# In[8]:X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
# In[9]:from sklearn.tree import DecisionTreeClassifier
# In[10]:dtree = DecisionTreeClassifier()
# In[11]:dtree.fit(X_train,y_train)
# In[12]:predictions = dtree.predict(X_test)
# In[13]:from sklearn.metrics import classification_report,confusion_matrix
# In[14]:
print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
# In[15]:from sklearn import tree
# In[16]:
with open("dtree.txt", "w") as f:
    f = tree.export_graphviz(dtree, out_file=f)
# In[17]:
with open("dtree.dot", "w") as f:
    f = tree.export_graphviz(dtree, out_file=f)
# In[18]:dot -Tpdf dtree.dot -o dtree.pdf
features = list (df119.columns[1:])
features
# In[ ]:features = list (X.columns[1:])
features
```

```
# In[ ]:import pydot
from IPython.display import Image
from sklearn.externals.six import StringIO
# In[ ]:
dot_data = StringIO()
export_graphviz(dtree, out_file=dot_data, feature_names=features, filled=True)
graph = pydot.graph_from_dot_data(dot_data.getvalue())
Image(graph[0].create_png())
```

Appendix D

Code: Support Vector Machine

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
from sklearn.model_selection import train_test_split
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
X1 = df119.drop('CALI', axis=1)
X = X1.drop('Fracture', axis=1)
y = X1['Fracture']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
from sklearn.svm import SVC
model = SVC()
model.fit(X_train,y_train)
predictions = model.predict(X_test)
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
from sklearn.model_selection import GridSearchCV
param_grid = {'C':[0.1,1,10,100,1000],'gamma':[1,0.1,0.01,0.001,0.0001]}
grid = GridSearchCV(SVC(),param_grid,verbose=3)
grid.fit(X_train,y_train)
grid.best_params_
grid.best_estimator_
grid_predictions = grid.predict(X_test)
print(confusion_matrix(y_test,grid_predictions))
print('\n')
print(classification_report(y_test,grid_predictions))
```

Appendix E

Code: Deep Learning

```
# -*- coding: utf-8 -*-
"""DeepLernWell.ipynb
Automatically generated by Colaboratory.
Original file is located at
    https://colab.research.google.com/drive/1zWaVcXduLeQwrcm8M5UPWRxTfcL37M3d
"""
# Commented out IPython magic to ensure Python compatibility.
import tensorflow as tf
import numpy as np
import pandas as pd
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
# %matplotlib inline
import pylab as plot
import matplotlib as mpl
import pandas.util.testing as tm
from google.colab import files
uploaded = files.upload()
import io
df318 = pd.read_excel(io.BytesIO(uploaded['GS#318.xlsx']))
df318
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X1 = df318.drop('CALI', axis=1)
X318 = X1.drop('Fracture', axis=1)
y318 = X1['Fracture']
scaler.fit(X318)
scaled_feature = scaler.fit_transform(X318)
df_feat = pd.DataFrame(scaled_feature,columns=X1.columns[:-1])
df_feat.head()
X = df_feat.drop('Fracture', axis = 1)
y = X1['Fracture']
from tensorflow import feature_column
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split
tf.keras.backend.floatx()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
def train_input_fn(X_train, y_train):
    dataset = tf.data.Dataset.from_tensor_slices((dict(X_train), y_train))
    dataset = dataset.shuffle(1000).repeat().batch(10)
    return dataset
def eval_input_fn(X_test, y_test):
    dataset = tf.data.Dataset.from_tensor_slices((dict(X_test), y_test))
    return dataset.shuffle(1000).repeat().batch(10)
```

```
my_feature_columns = []
for key in (X_train).keys():
 my_feature_columns.append(tf.feature_column.numeric_column(key=key))
optimizer_adam= tf.optimizers.Adam(learning_rate=0.01)
model=tf.estimator.DNNClassifier( feature_columns=my_feature_columns, hidden_units=[100,100, 100],
optimizer=optimizer_adam, n_classes=2, model_dir='models/iris')
model.train(input_fn=lambda:train_input_fn(X_train, y_train), steps=200)
eval_results3 = model.evaluate(input_fn=lambda:eval_input_fn(X_test, y_test), steps=1)
eval_results3
eval_results2
eval_results
print('AUC: {}'.format(eval_results['auc']))
print('AUC: {}'.format(eval_results2['auc']))
print('AUC: {}'.format(eval_results3['auc']))
```

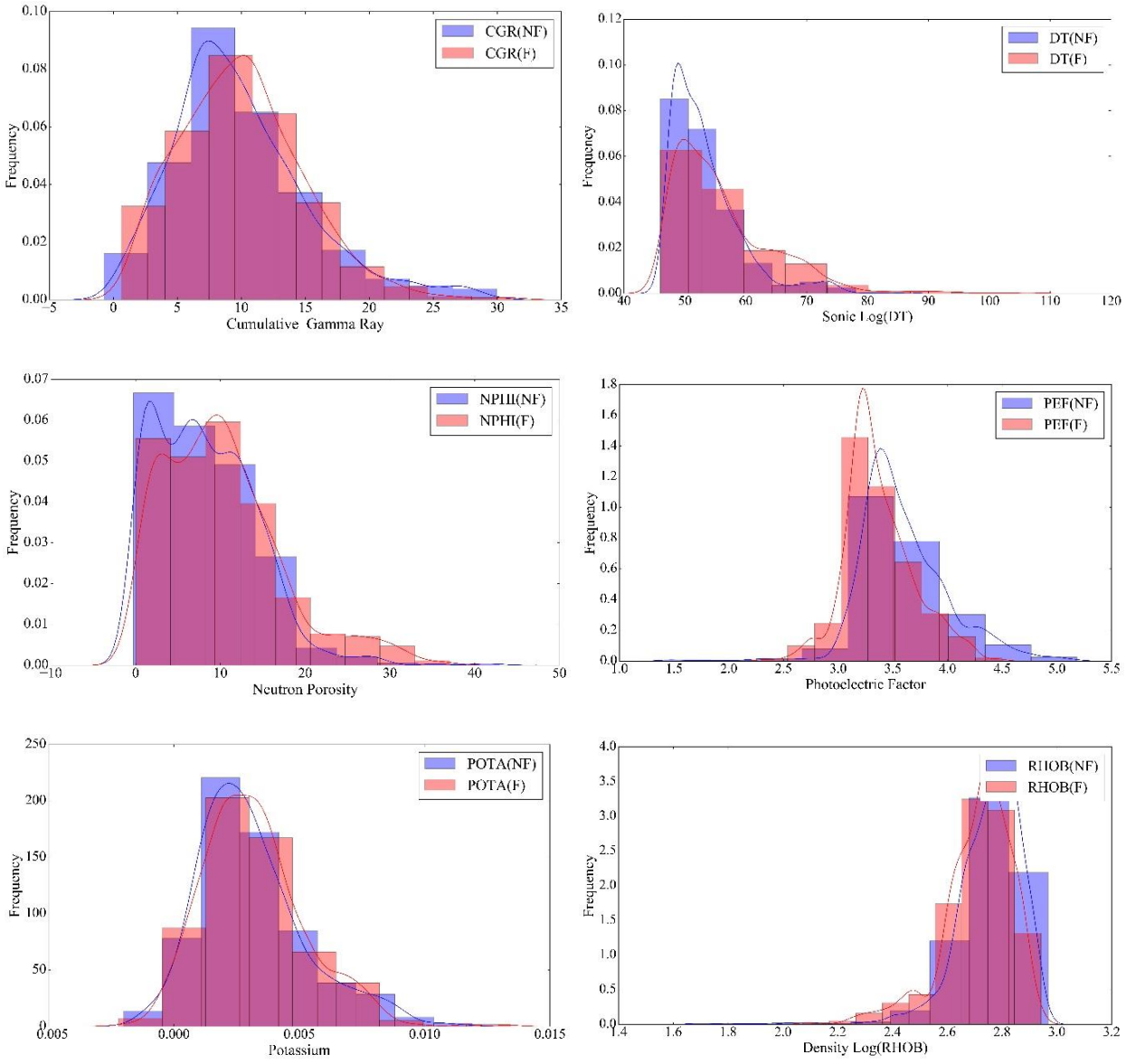
Appendix F

Code: Correlation Coefficient

```
#!/usr/bin/env python
# coding: utf-8
pd.read_excel('GS#119.xls')
import pandas as pd
pd.read_excel('GS#119.xls')
import xlsxwriter
import pandas as pd
from pandas import DataFrame
path = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
xl = pd.ExcelFile(path)
print(xl.sheet_names)
df1 = xl.parse('Sheet1')
df1.count()
df1.groupby('group').apply(lambda d: d.filter(like=df1.drop('Fracture', axis = 1)).corrwith(df1.['Fracture']))
df2 = df1.drop('Fracture', axis = 1)
df3 = df1['Fracture']
df1.apply(lambda d: d.filter(like=df1.drop('Fracture', axis = 1)).corrwith(df1['Fracture']))
corr = pg.pairwise_corr(df1, columns=[['Fracture'], df2], method='pearson')
import pingouin as pg
corrMatrix = df1.corr()
corrMatrix
# In[ ]:
```

Appendix G

Histogram and corresponding PDF for different logs in fractured zones and non-fractured ones in wells 2 and 5



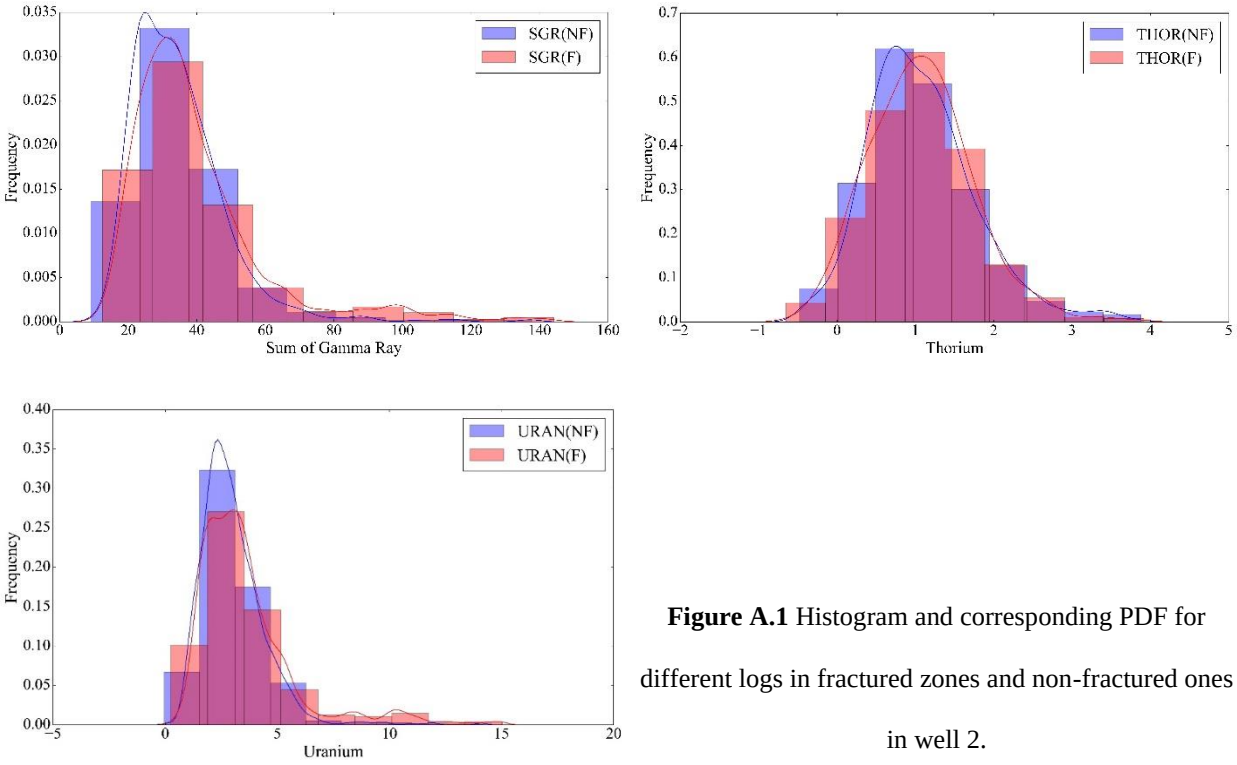
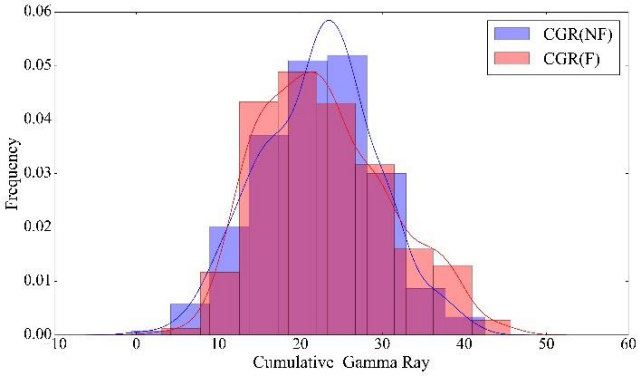
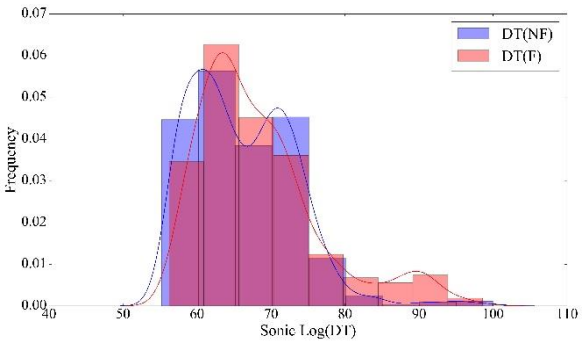
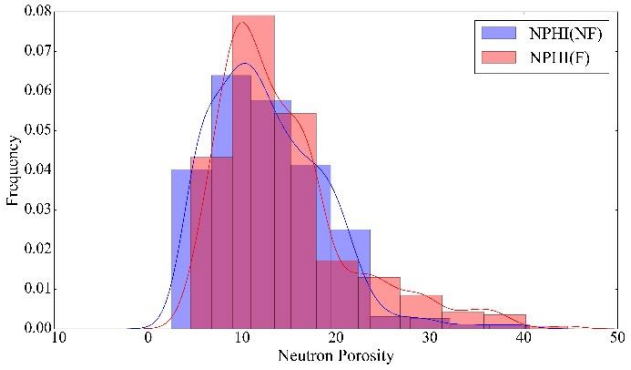
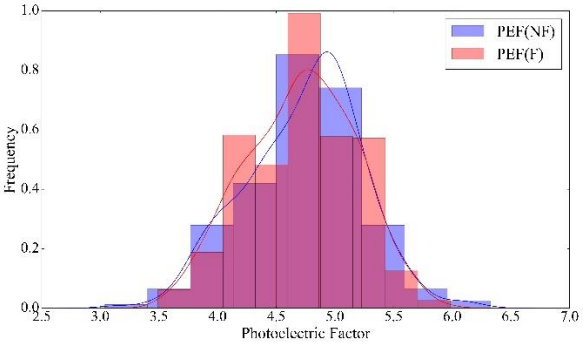


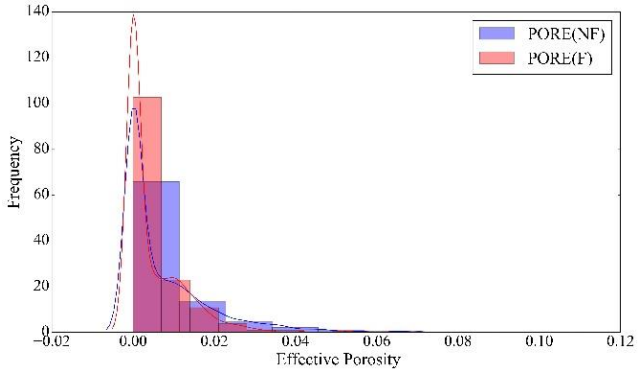
Figure A.1 Histogram and corresponding PDF for different logs in fractured zones and non-fractured ones in well 2.

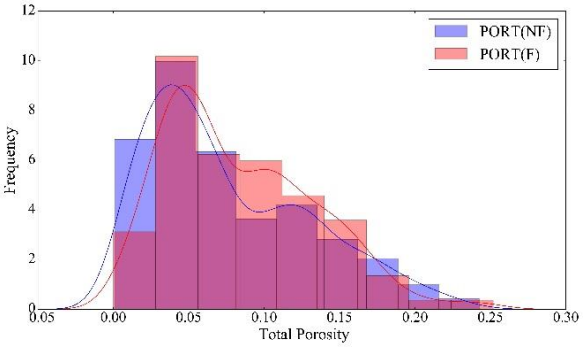

CGR(NF)
CGR(F)
Frequency
Cumulative Gamma Ray

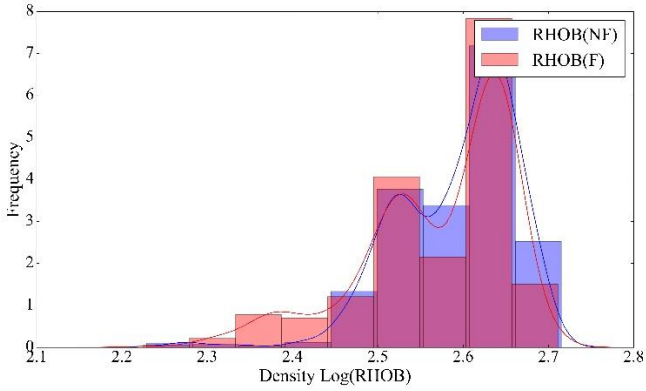

DT(NF)
DT(F)
Frequency
Sonic Log(DT)

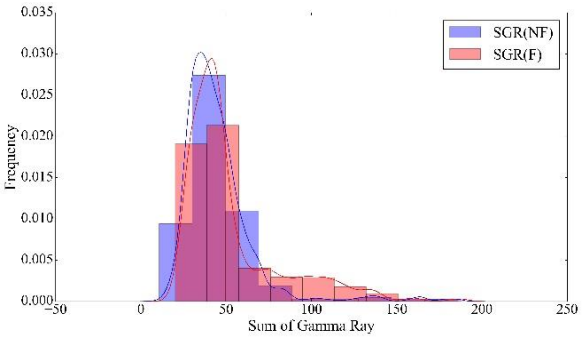

NPHI(NF)
NPHI(F)
Frequency
Neutron Porosity


PEF(NF)
PEF(F)
Frequency
Photoelectric Factor


PORE(NF)
PORE(F)
Frequency
Effective Porosity


PORT(NF)
PORT(F)
Frequency
Total Porosity


RHOB(NF)
RHOB(F)
Frequency
Density Log(RHOB)


SGR(NF)
SGR(F)
Frequency
Sum of Gamma Ray

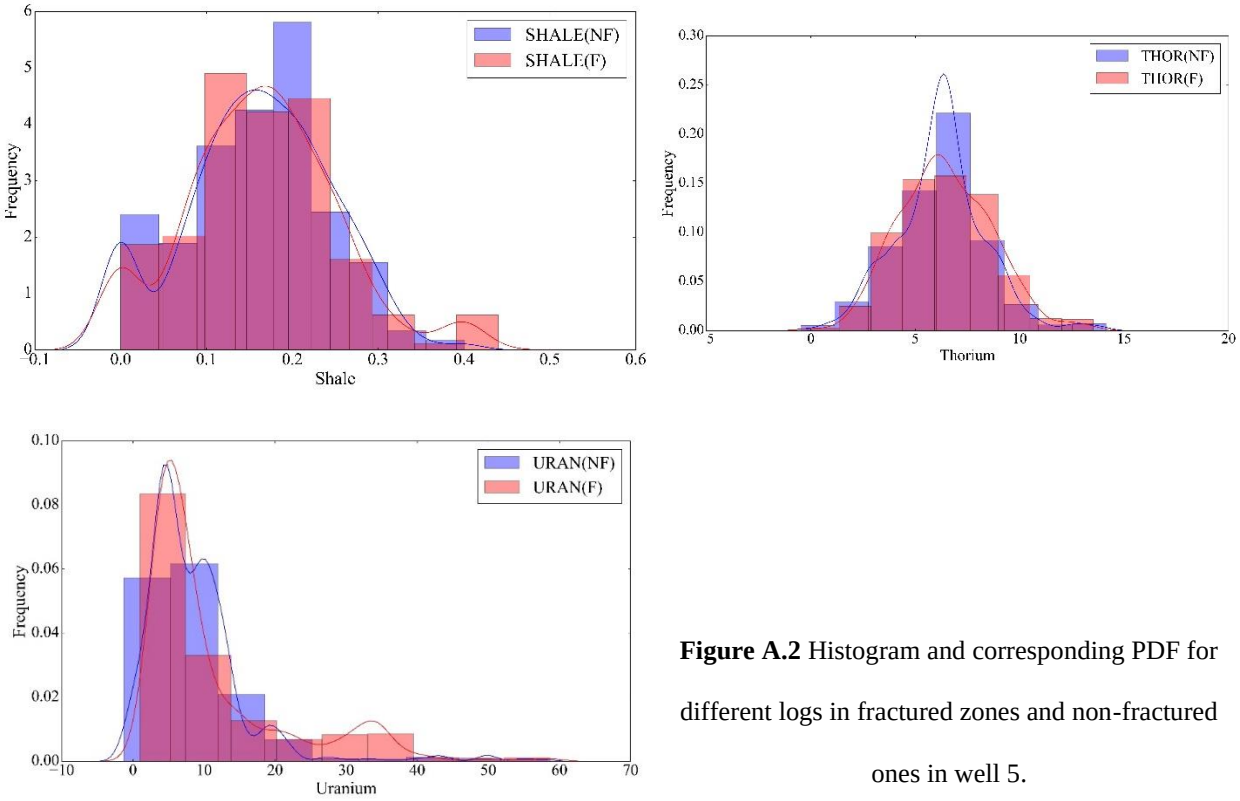


Figure A.2 Histogram and corresponding PDF for different logs in fractured zones and non-fractured ones in well 5.

Appendix H

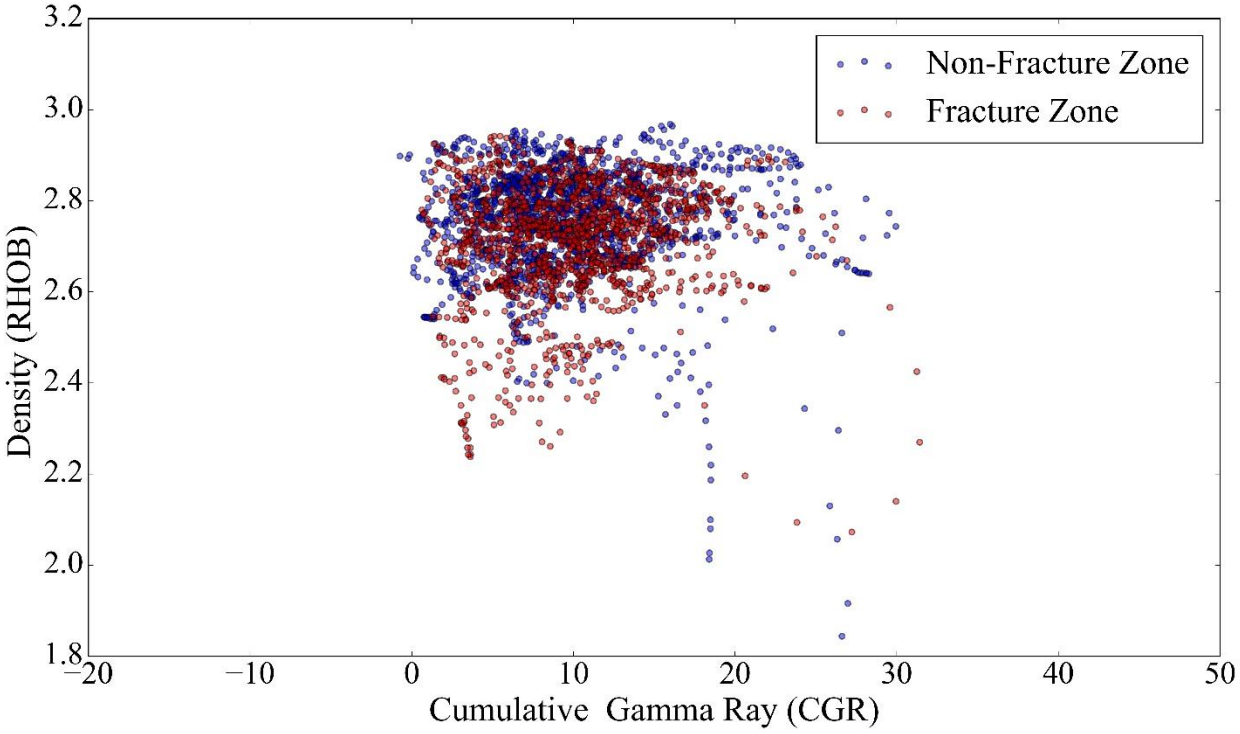
Code: Histogram and corresponding PDF for different logs in fractured zones and non-fractured ones

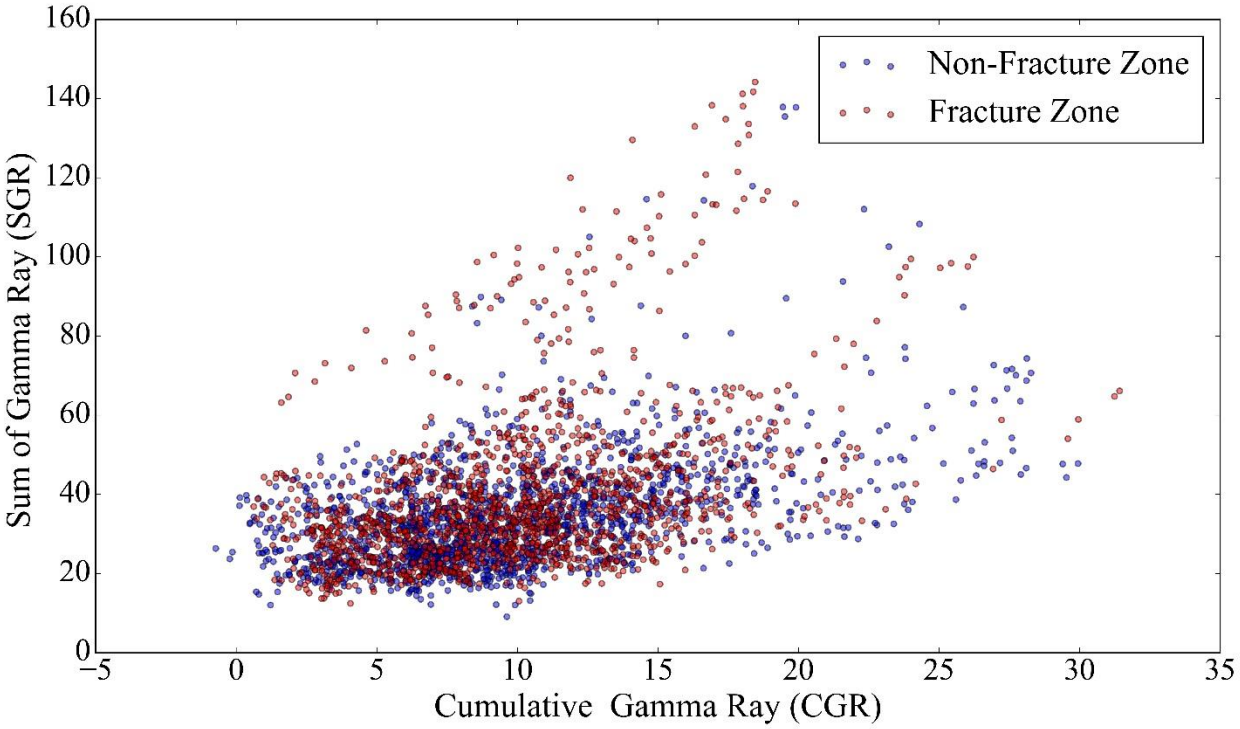
```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
df119a = df119.drop('Fracture', axis = 1)
df119Logs = df119a.drop('DEPTH', axis = 1)
X = df119Logs.iloc[1:1216, :]
Y = df119Logs.iloc[1216::, :]
plt.hist(X['CALI'], bins=10, alpha = 0.5, color = 'r')
plt.figure(figsize=(16,9))
try:
    sns.distplot(X["CALI"], bins=9, label="CALI(NF)")
    sns.distplot(Y["CALI"], bins=9, label="CALI(F)")
except RuntimeError as re:
    if str(re).startswith("Selected KDE bandwidth is 0. Cannot estimate density."):
        sns.distplot(X['CALI'], kde_kws={'bw': 0.1})
    else:
        raise re
plt.legend()
import pylab as plot
import matplotlib as mpl
mpl.rc('font',family='Times New Roman')
params = {'legend.fontsize': 27,
          'legend.handlelength': 2
          }
plot.rcParams.update(params)
CGR_PDF = plt.figure(figsize=(16,9))
try:
    sns.distplot(X["CGR"],color="b", bins=9, label="CGR(NF)")
    sns.distplot(Y["CGR"],color="r", bins=9, label="CGR(F)")
except RuntimeError as re:
    if str(re).startswith("Selected KDE bandwidth is 0. Cannot estimate density."):
        sns.distplot(X['CGR'], kde_kws={'bw': 0.1})
    else:
        raise re
plt.legend()
plt.xlabel('Cumulative  Gamma Ray', size = 27, fontname="Times New Roman")
```

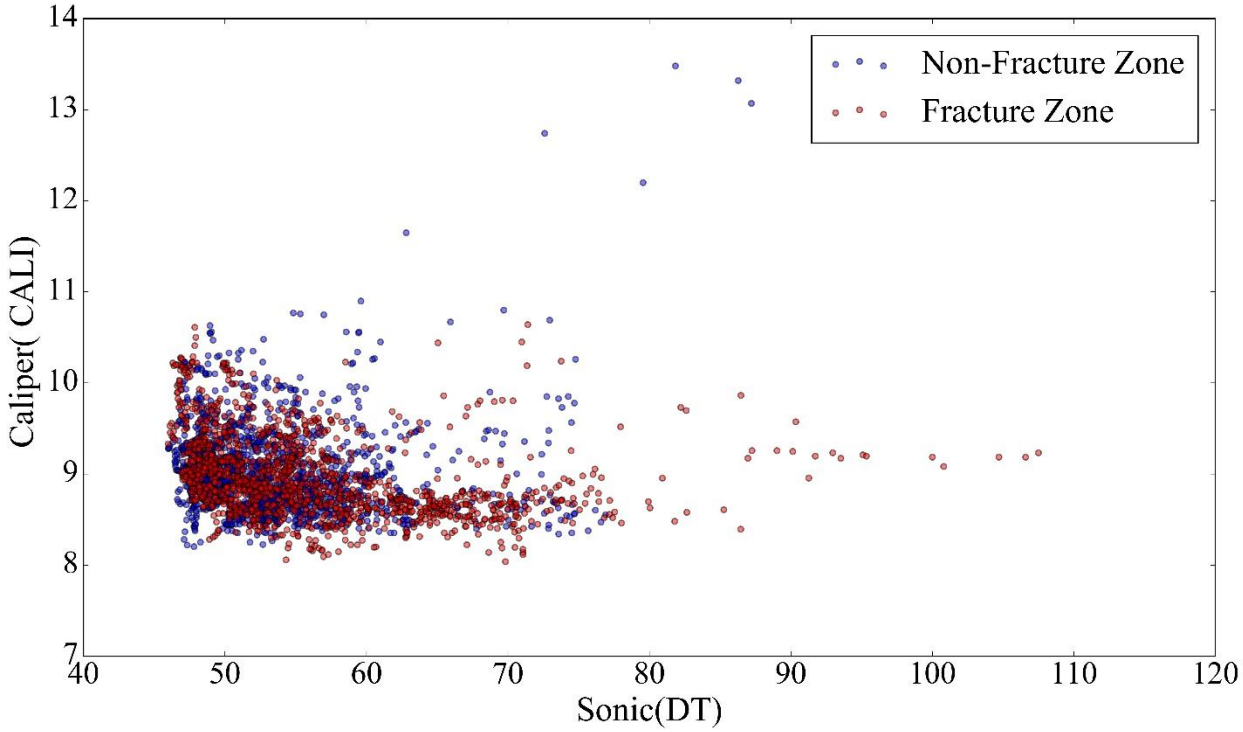
```
plt.ylabel('Frequency', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
```

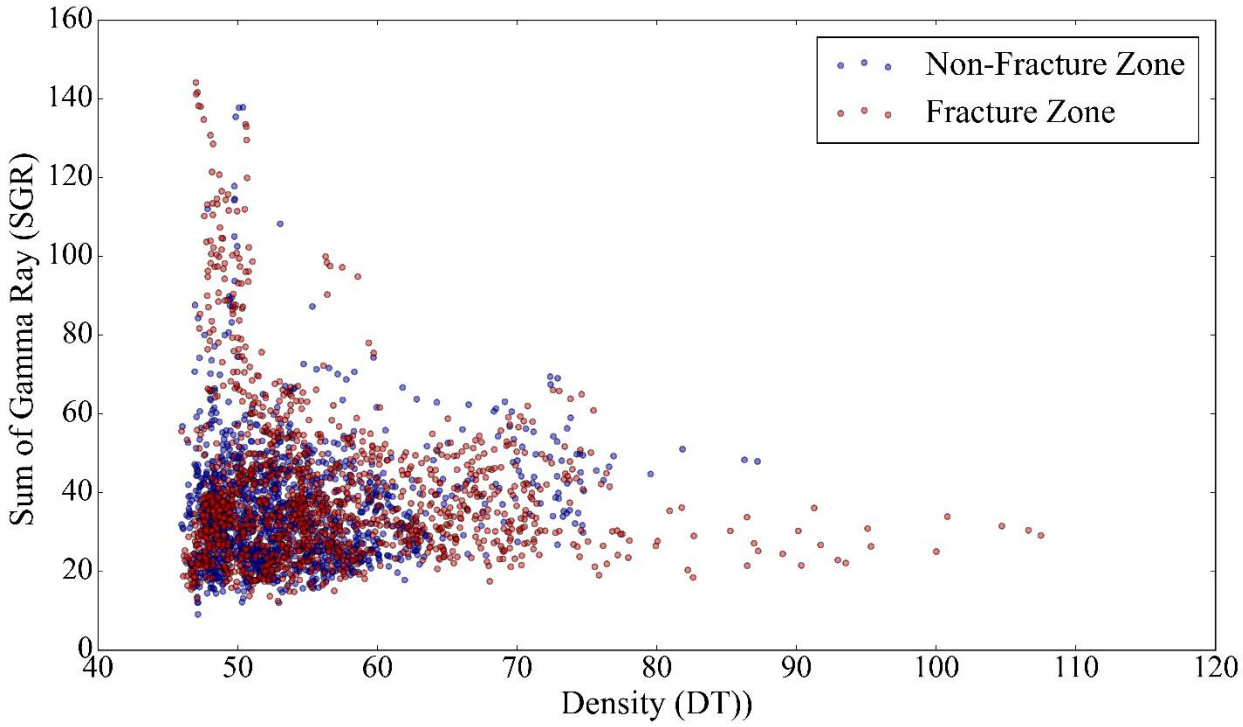
Appendix I

Cross plots of selected petrophysical logs in fractured and non-fractured zones in wells 2 and 5






Non-Fracture Zone
Fracture Zone
Caliper(CALI)
Sonic(DT)
14
13
12
11
10
9
8
7
40
50
60
70
80
90
100
110
120


Non-Fracture Zone
Fracture Zone
Sum of Gamma Ray (SGR)
Density (DT))
160
140
120
100
80
60
40
20
0
40
50
60
70
80
90
100
110
120

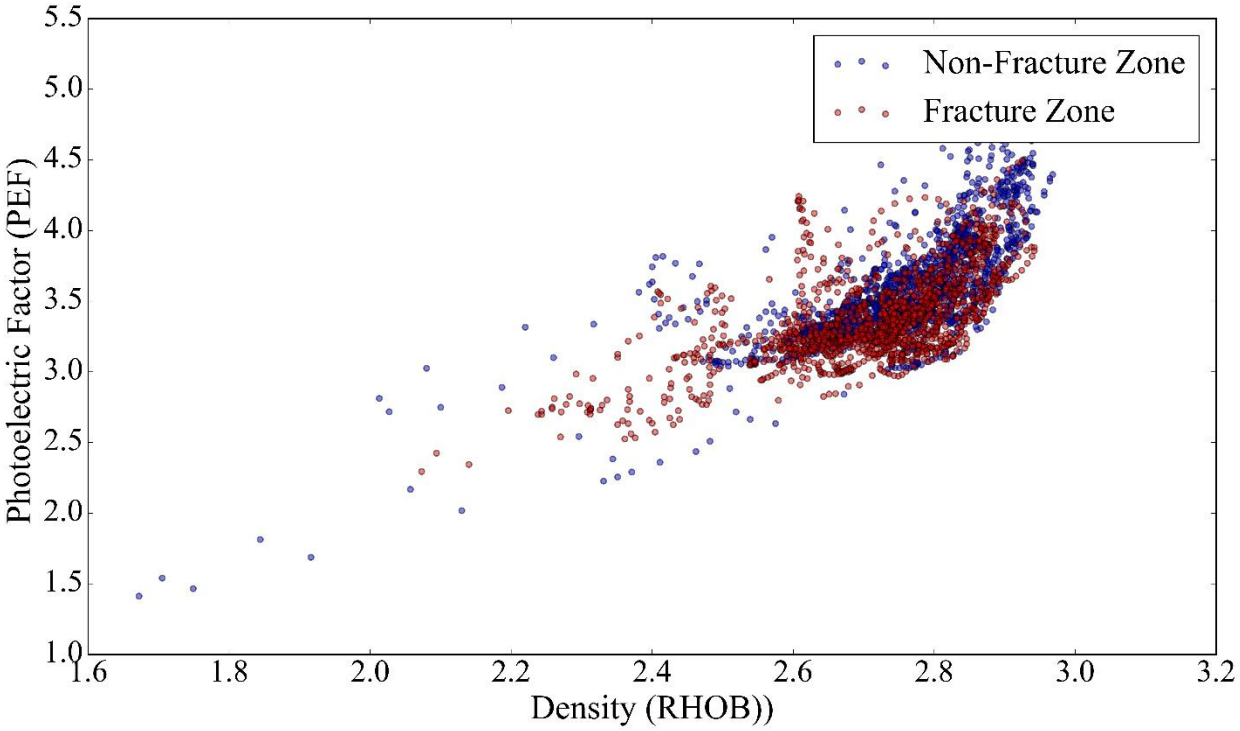
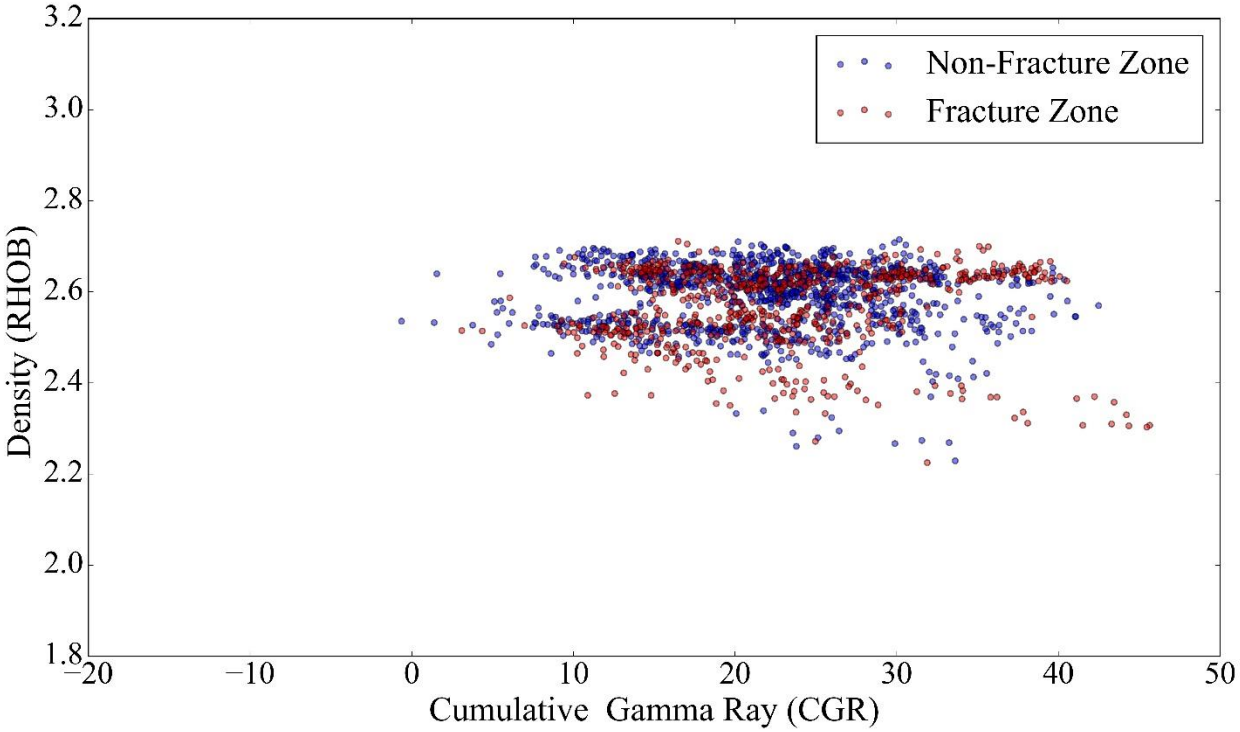
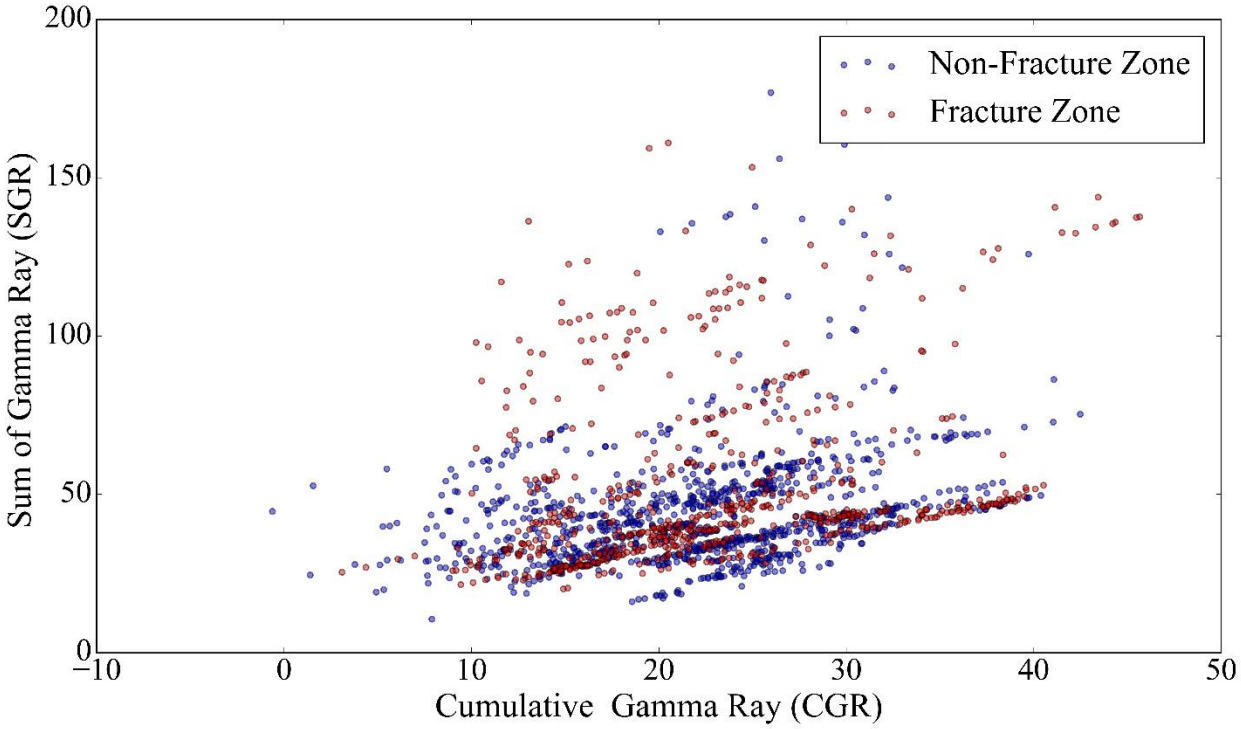
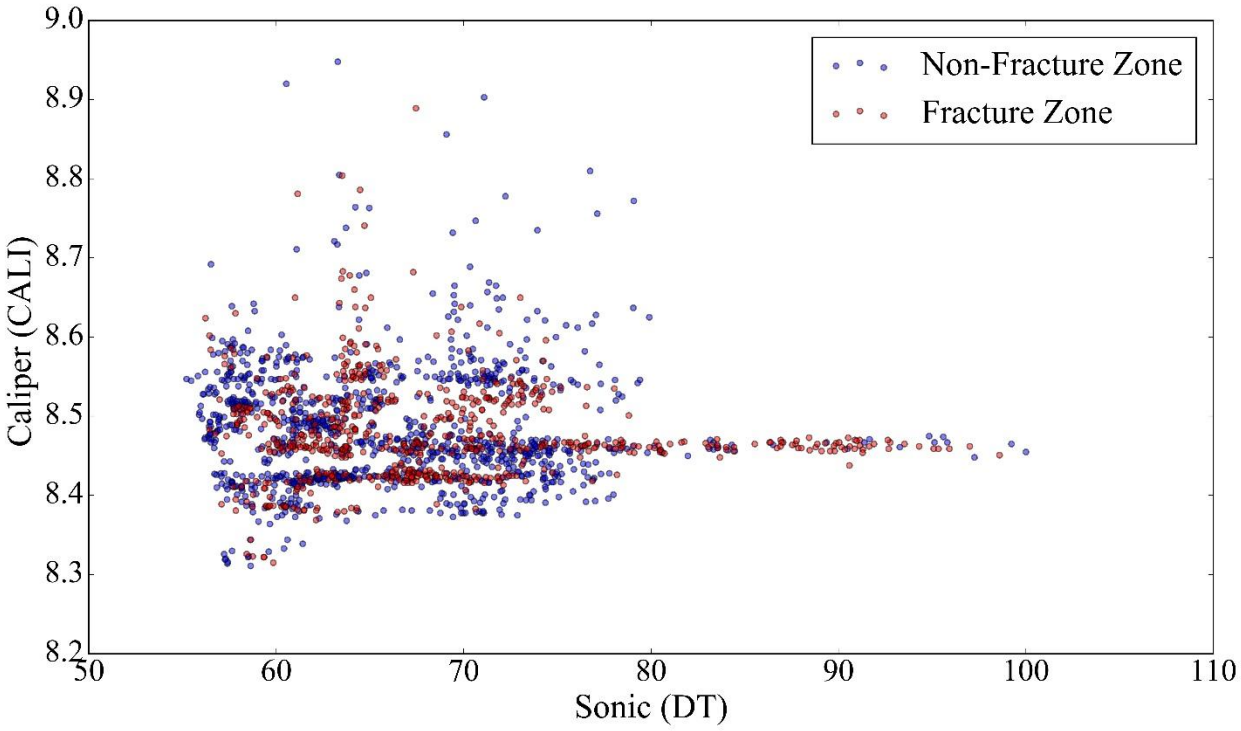
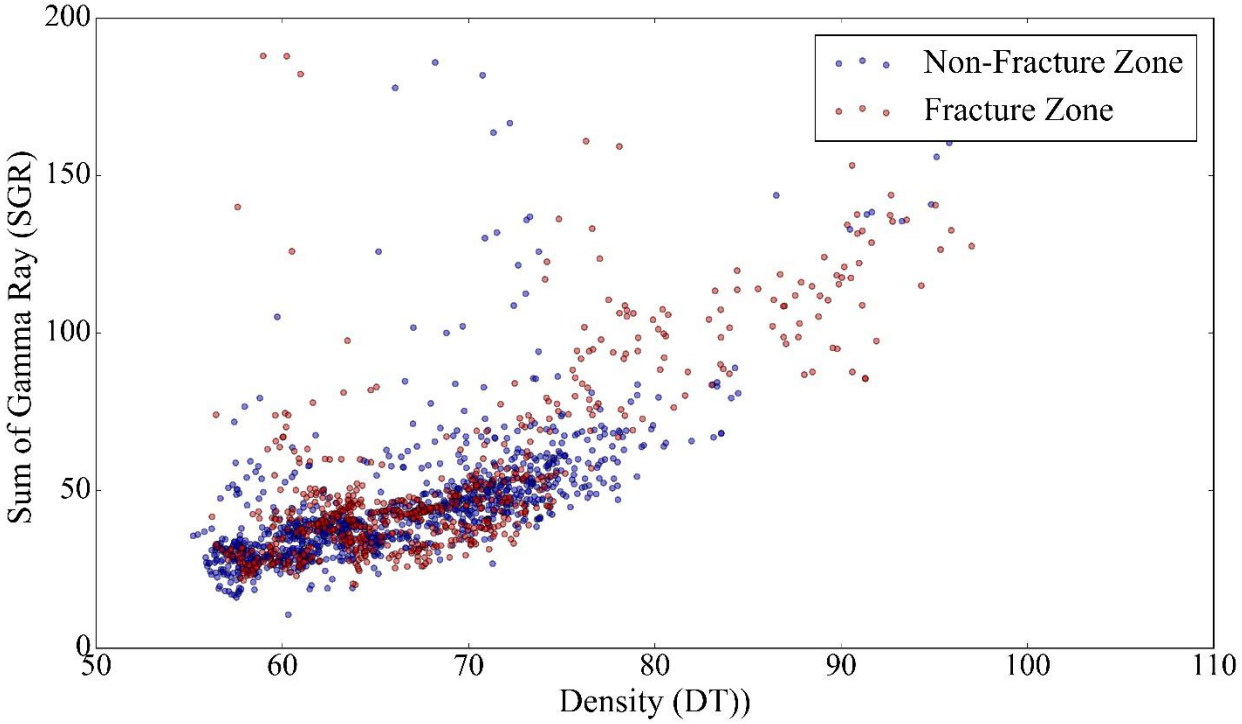


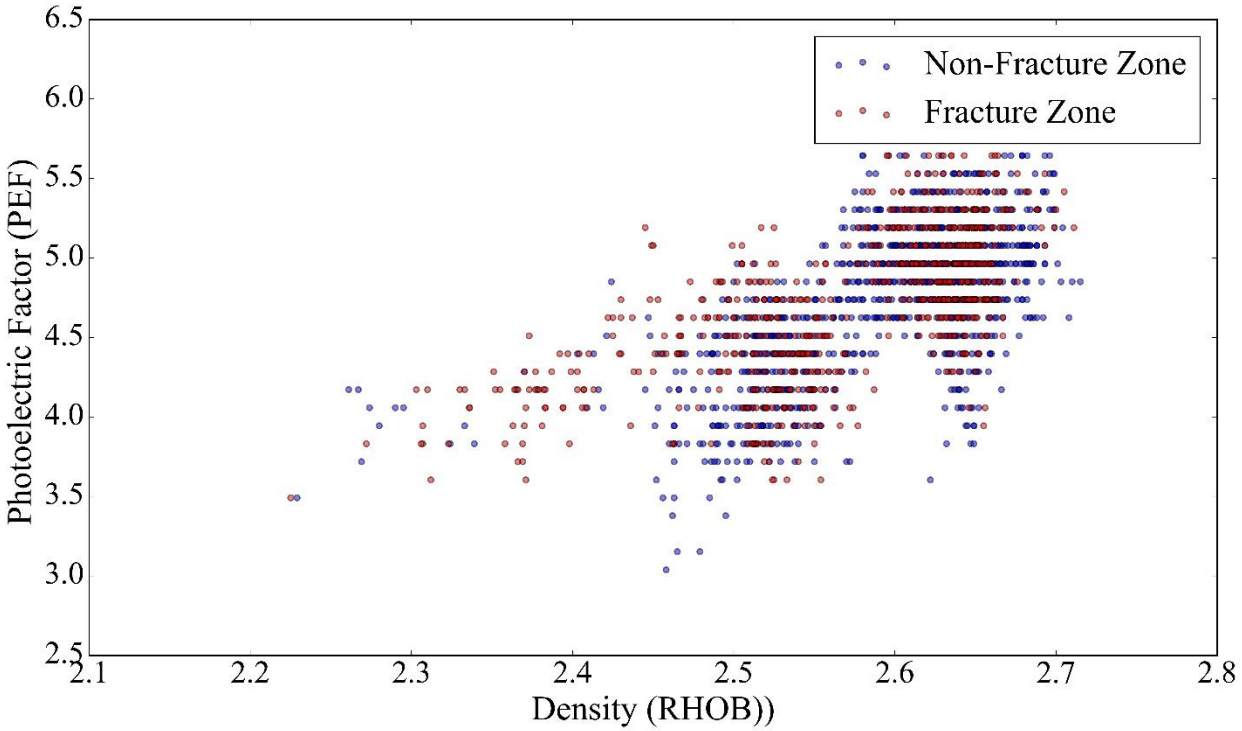
Figure B.1 Cross plots of selected petrophysical logs in fractured and non-fractured zones in well 2.




Non-Fracture Zone
Fracture Zone
Sum of Gamma Ray (SGR)
Cumulative Gamma Ray (CGR)
200
150
100
50
0
−10
0
10
20
30
40
50


Non-Fracture Zone
Fracture Zone
Caliper (CALI)
Sonic (DT)
9.0
8.9
8.8
8.7
8.6
8.5
8.4
8.3
8.2
50
60
70
80
90
100
110


Non-Fracture Zone
Fracture Zone
Sum of Gamma Ray (SGR)
Density (DT))
0
50
100
150
200
50
60
70
80
90
100
110


Non-Fracture Zone
Fracture Zone
Photoelectric Factor (PEF)
Density (RHOB))
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8

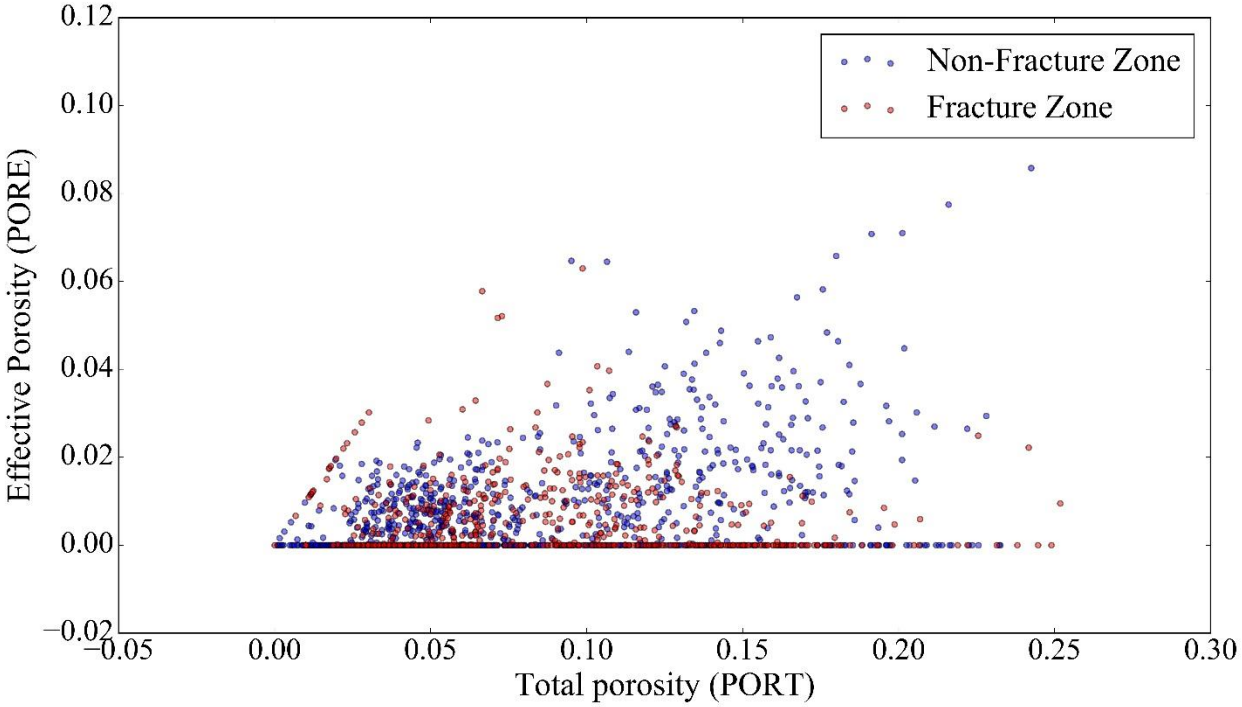


Figure B.2 Cross plots of selected petrophysical logs in fractured and non-fractured zones in well 5.

Appendix J

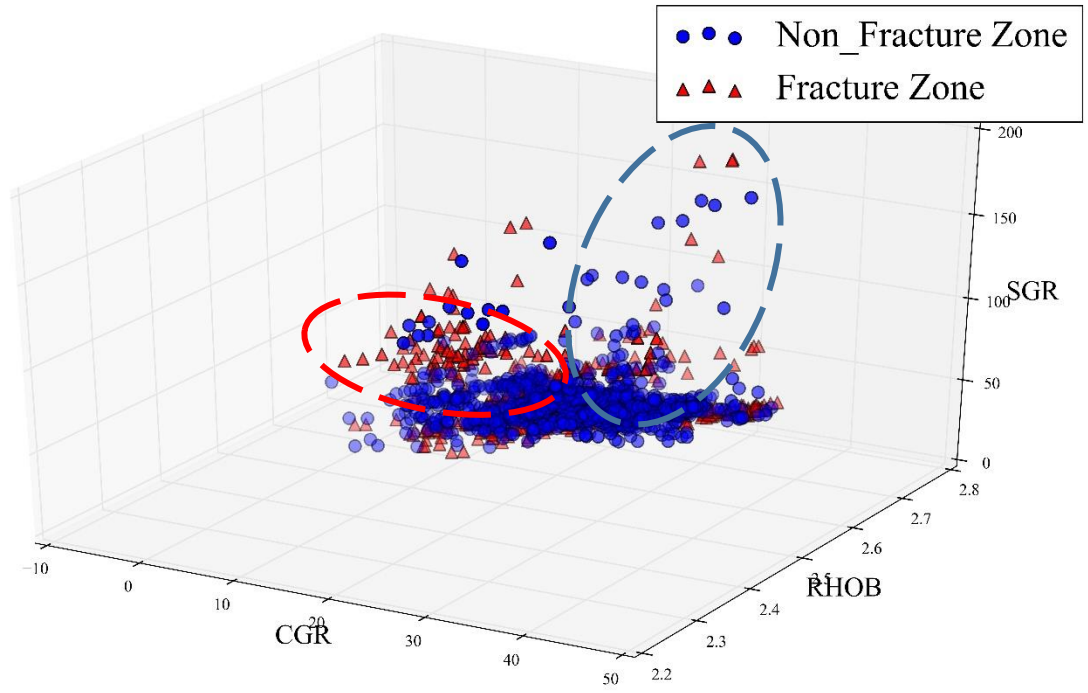
Code: 2D Cross plots of selected petrophysical logs in fractured and non-fractured zones in wells 2

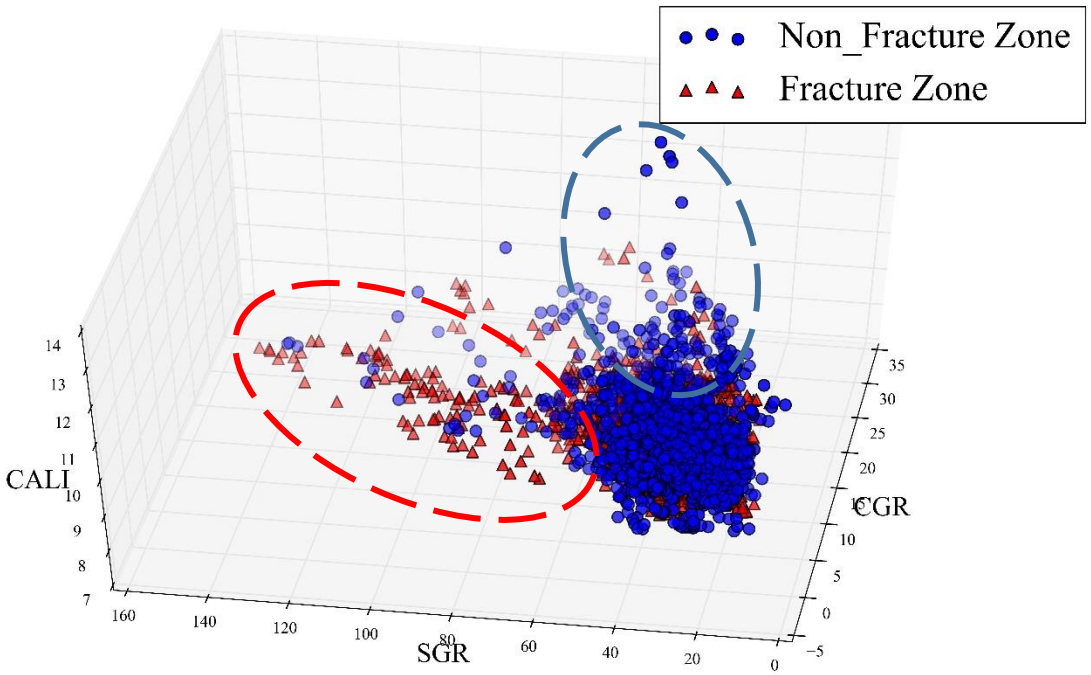
```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
df119a = df119.drop('Fracture', axis = 1)
df119Logs = df119a.drop('DEPTH', axis = 1)
df119NoF = df119Logs.iloc[1:1216, :]
df119F = df119Logs.iloc[1216::, :]
CGR_RHOB119 = plt.figure(figsize=(16,9))
df119NoF['CGR'].plot()
df119NoF['RHOB'].plot()
df119F['CGR'].plot()
df119F['RHOB'].plot()
import pylab as plot
import matplotlib as mpl
mpl.rc('font',family='Times New Roman')
params = {'legend.fontsize': 27,
          'legend.handlelength': 2
          }
plot.rcParams.update(params)
CGR_RHOB119 = plt.figure(figsize=(16,9))
NF = plt.scatter(df119NoF['CGR'], df119NoF['RHOB'],c='blue', alpha=0.5)
F =  plt.scatter(df119F['CGR'], df119F['RHOB'],c='red', alpha=0.5)
import matplotlib.pyplot as plt
plt.xlim(-20,50)
plt.ylim(1.8,3.2)
plt.legend([NF, F], ['Non-Fracture Zone','Fracture Zone'])
plt.xlabel('Cumulative  Gamma Ray (CGR)', size = 27, fontname="Times New Roman")
plt.ylabel('Density (RHOB) ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
CGR_SGR119 = plt.figure(figsize=(16,9))
NF = plt.scatter(df119NoF['CGR'], df119NoF['SGR'],c='blue', alpha=0.5)
F =  plt.scatter(df119F['CGR'], df119F['SGR'],c='red', alpha=0.5)
import matplotlib.pyplot as plt
```

```
plt.legend([NF, F], ['Non-Fracture Zone','Fracture Zone'])
plt.xlabel('Cumulative  Gamma Ray (CGR)', size = 27, fontname="Times New Roman")
plt.ylabel('Sum of Gamma Ray (SGR) ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
NPHI_SW119 = plt.figure(figsize=(16,9))
NF = plt.scatter(df119NoF['NPHI'], df119NoF['SW'],c='blue', alpha=0.5)
F =  plt.scatter(df119F['NPHI'], df119F['SW'],c='red', alpha=0.5)
import matplotlib.pyplot as plt
plt.legend([NF, F], ['Non-Fracture Zone','Fracture Zone'])
plt.xlabel('Neutron Porosity (NPHI)', size = 27, fontname="Times New Roman")
plt.ylabel('Water Saturation (SW) ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
DT_CALI119 = plt.figure(figsize=(16,9))
NF = plt.scatter(df119NoF['DT'], df119NoF['CALI'],c='blue', alpha=0.5)
F =  plt.scatter(df119F['DT'], df119F['CALI'],c='red', alpha=0.5)
import matplotlib.pyplot as plt
plt.legend([NF, F], ['Non-Fracture Zone','Fracture Zone'])
plt.xlabel('Sonic (DT)', size = 27, fontname="Times New Roman")
plt.ylabel('Caliper (CALI) ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
RHOB_PEF119 = plt.figure(figsize=(16,9))
NF = plt.scatter(df119NoF['RHOB'], df119NoF['PEF'],c='blue', alpha=0.5)
F =  plt.scatter(df119F['RHOB'], df119F['PEF'],c='red', alpha=0.5)
import matplotlib.pyplot as plt
plt.legend([NF, F], ['Non-Fracture Zone','Fracture Zone'])
plt.xlabel('Density (RHOB))', size = 27, fontname="Times New Roman")
plt.ylabel('Photoelectric Factor (PEF) ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
DT_SGR119 = plt.figure(figsize=(16,9))
NF = plt.scatter(df119NoF['DT'], df119NoF['SGR'],c='blue', alpha=0.5)
F =  plt.scatter(df119F['DT'], df119F['SGR'],c='red', alpha=0.5)
import matplotlib.pyplot as plt
plt.legend([NF, F], ['Non-Fracture Zone','Fracture Zone'])
plt.xlabel('Density (DT))', size = 27, fontname="Times New Roman")
plt.ylabel('Sum of Gamma Ray (SGR) ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
plt.yticks(size = 25, fontname="Times New Roman")
CGR_RHOB119.savefig("CGR_RHOB119.JPEG", bbox_inches='tight', dpi=600)
CGR_SGR119.savefig("CGR_SGR119.JPEG", bbox_inches='tight', dpi=600)
NPHI_SW119.savefig("NPHI_SW119.JPEG", bbox_inches='tight', dpi=600)
DT_CALI119.savefig("DT_CALI119.JPEG", bbox_inches='tight', dpi=600)
RHOB_PEF119.savefig("RHOB_PEF119.JPEG", bbox_inches='tight', dpi=600)
DT_SGR119.savefig("DT_SGR119.JPEG", bbox_inches='tight', dpi=600)
path245 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
```

Appendix K

3D cross plots of selected petrophysical logs in fractured/non-fractured zones in wells 2 and 5





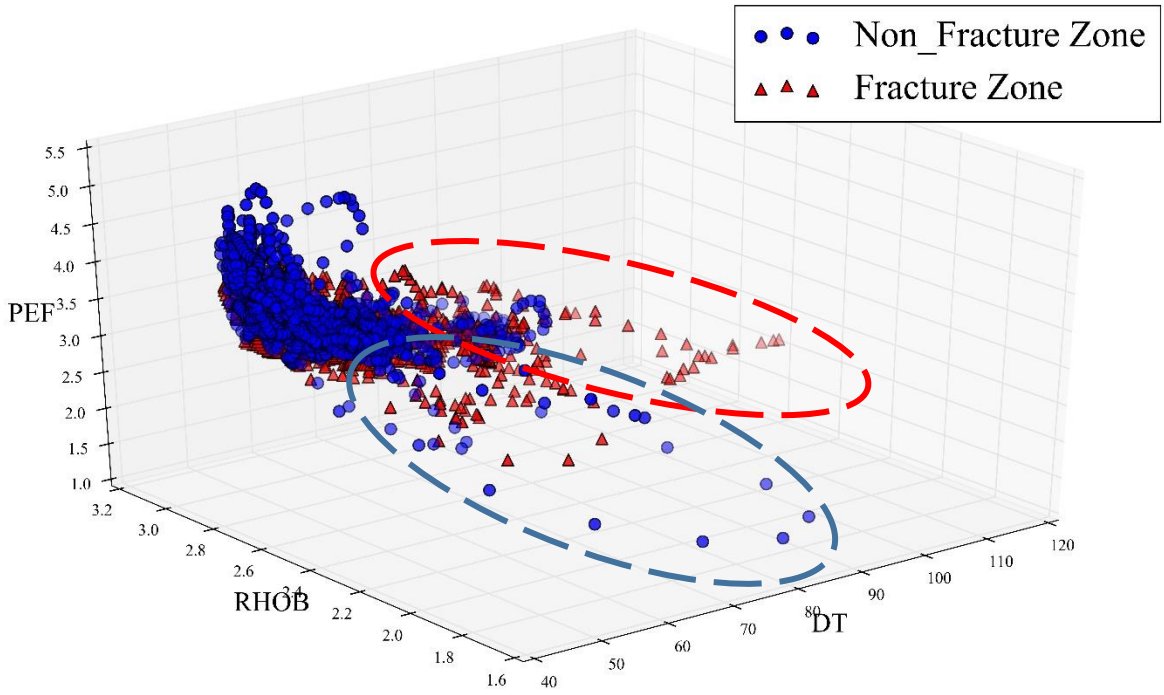
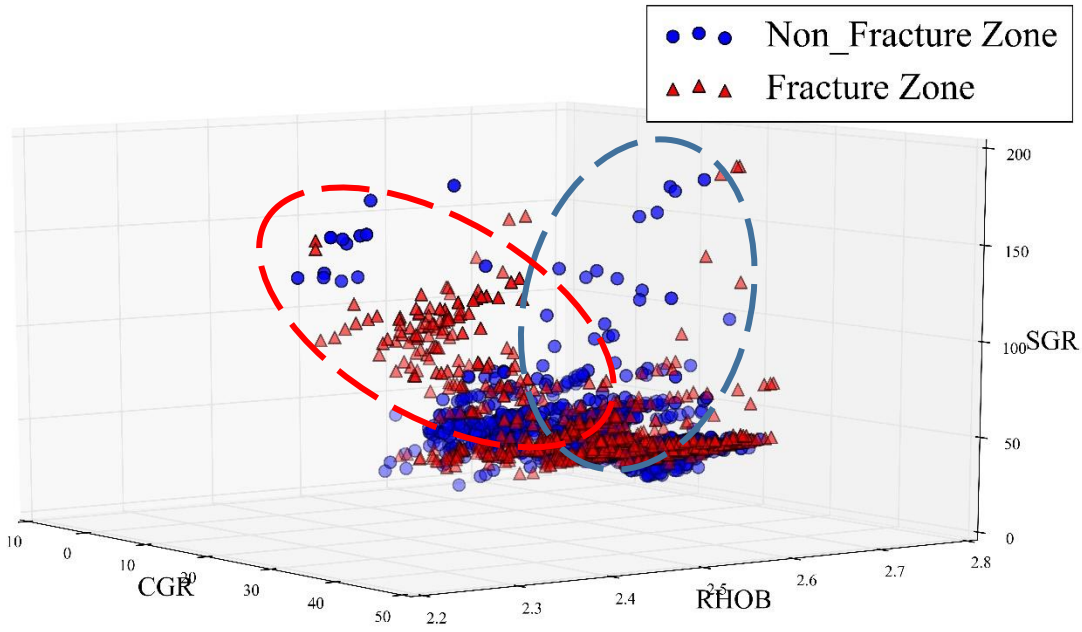
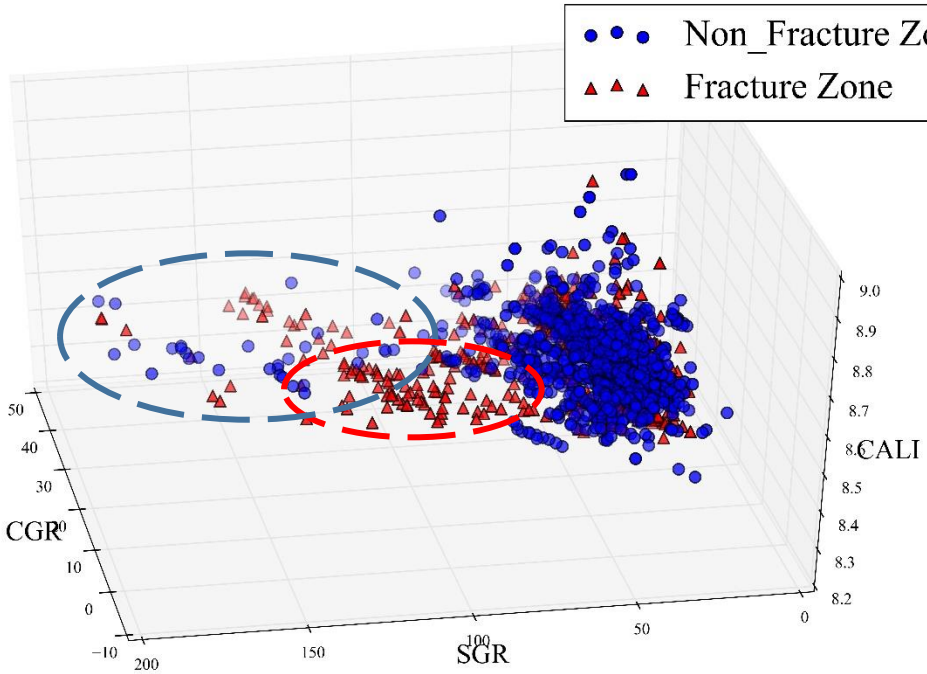


Figure C.1 3D cross plots of selected petrophysical logs in order to show how higher feature space could help to discriminate fractured zones from non-fractured ones in well 2.


Non_Fracture Zone
Fracture Zone
CGR
RHOB
SGR


Non_Fracture Zone
Fracture Zone
CGR
SGR
CALI

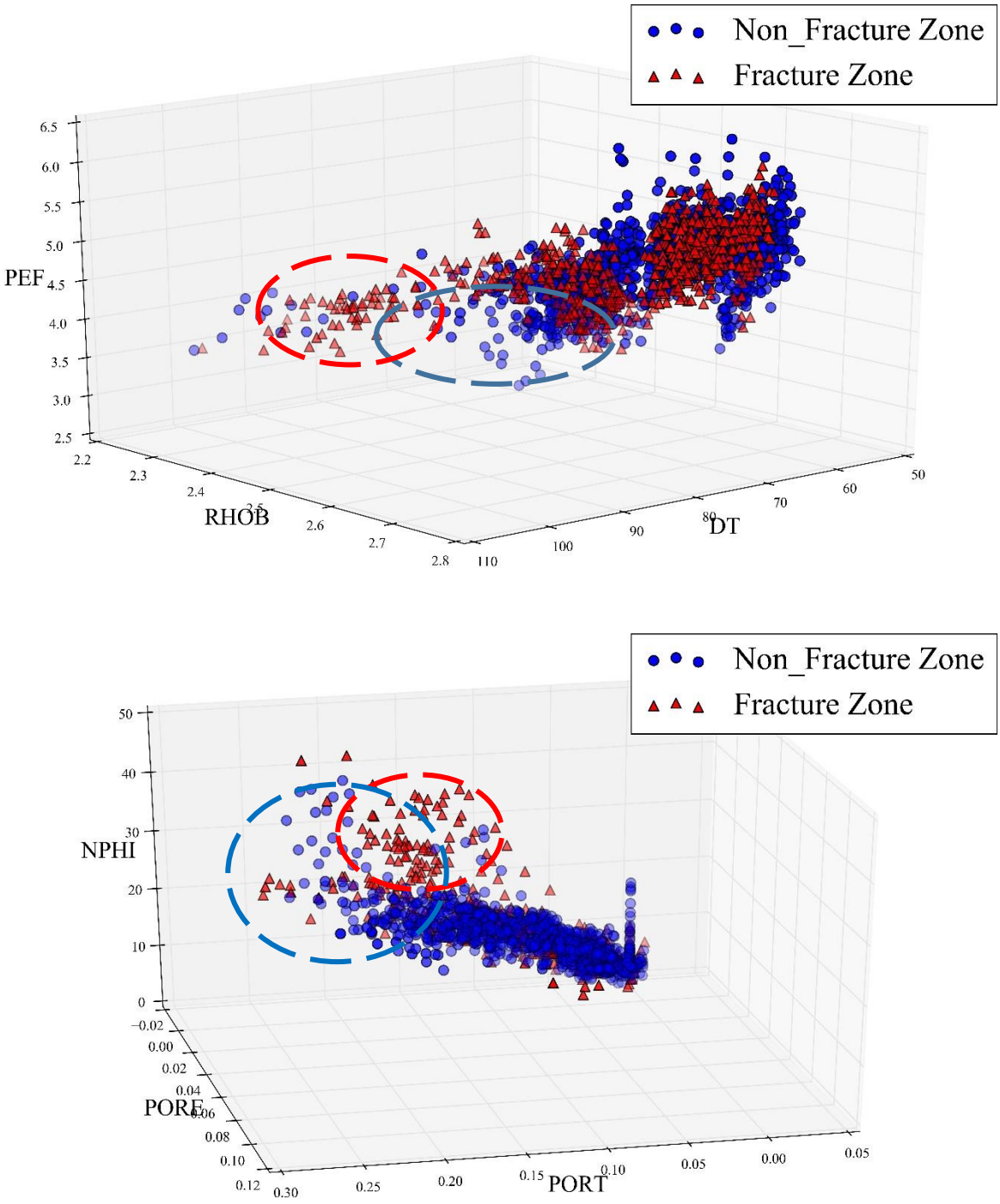


Figure C.2 3D cross plots of selected petrophysical logs in order to show how higher feature space could help to discriminate fractured zones from non-fractured ones in well 5.

Appendix L

Code: 3D cross plots of selected petrophysical logs in fractured/non-fractured zones in wells 2

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x = pd.ExcelFile(path119)
df119 = x.parse('Sheet1')
df119a = df119.drop('Fracture', axis = 1)
df119Logs = df119a.drop('DEPTH', axis = 1)
df119NF = df119Logs.iloc[1:1216, :]
df119F = df119Logs.iloc[1216::, :]
from mpl_toolkits.mplot3d import axes3d
get_ipython().run_line_magic('matplotlib', 'notebook')
XF=df119F['CGR']
YF=df119F['RHOB']
ZF=df119F['SGR']
XNF=df119NF['CGR']
YNF=df119NF['RHOB']
ZNF=df119NF['SGR']
import pylab as plot
import matplotlib as mpl
mpl.rc('font',family='Times New Roman')
params = {'legend.fontsize': 27,
          'legend.handlelength': 2
          }
plot.rcParams.update(params)
CGR_RHOB_SGR119 = plt.figure(figsize=(16,9))
ax = CGR_RHOB_SGR119.gca(projection='3d')
scatter = ax.scatter3D(XNF, YNF, ZNF,s=80, c='blue',cmap=plt.cm.viridis, marker='o', label='Non_Fracture
Zone')
scatter = ax.scatter3D(XF, YF, ZF,s=80, c='red',cmap=plt.cm.viridis, marker='^', label='Fracture Zone')
plt.show()
ax.legend()
ax.set_xlabel('CGR', size = 20, fontname="Times New Roman")
ax.set_ylabel('RHOB', size = 20, fontname="Times New Roman")
ax.set_zlabel('SGR', size = 20, fontname="Times New Roman")
XF=df119F['CGR']
YF=df119F['SGR']
```

```
ZF=df119F['CALI']
XNF=df119NF['CGR']
YNF=df119NF['SGR']
ZNF=df119NF['CALI']
CGR_SGR_CALI119 = plt.figure(figsize=(16,9))
ax = CGR_SGR_CALI119.gca(projection='3d')
scatter = ax.scatter3D(XNF, YNF, ZNF,s=80, c='blue',cmap=plt.cm.viridis, marker='o', label='Non_Fracture
Zone')
scatter = ax.scatter3D(XF, YF, ZF,s=80, c='red',cmap=plt.cm.viridis, marker='^', label='Fracture Zone')
plt.show()
ax.legend()
ax.set_xlabel('CGR', size = 20, fontname="Times New Roman")
ax.set_ylabel('SGR', size = 20, fontname="Times New Roman")
ax.set_zlabel('CALI', size = 20, fontname="Times New Roman")
XF=df119F['DT']
YF=df119F['RHOB']
ZF=df119F['PEF']
XNF=df119NF['DT']
YNF=df119NF['RHOB']
ZNF=df119NF['PEF']
DT_RHOB_PEF119 = plt.figure(figsize=(16,9))
ax = DT_RHOB_PEF119.gca(projection='3d')
scatter = ax.scatter3D(XNF, YNF, ZNF,s=80, c='blue',cmap=plt.cm.viridis, marker='o', label='Non_Fracture
Zone')
scatter = ax.scatter3D(XF, YF, ZF,s=80, c='red',cmap=plt.cm.viridis, marker='^', label='Fracture Zone')
plt.show()
ax.legend()
ax.set_xlabel('DT', size = 20, fontname="Times New Roman")
ax.set_ylabel('RHOB', size = 20, fontname="Times New Roman")
ax.set_zlabel('PEF', size = 20, fontname="Times New Roman")
XF=df119F['DT']
YF=df119F['SGR']
ZF=df119F['SW']
XNF=df119NF['DT']
YNF=df119NF['SGR']
ZNF=df119NF['SW']
DT_SGR_SW119 = plt.figure(figsize=(16,9))
ax = DT_SGR_SW119.gca(projection='3d')
scatter = ax.scatter3D(XNF, YNF, ZNF,s=80, c='blue',cmap=plt.cm.viridis, marker='o', label='Non_Fracture
Zone')
scatter = ax.scatter3D(XF, YF, ZF,s=80, c='red',cmap=plt.cm.viridis, marker='^', label='Fracture Zone')
plt.show()
ax.legend()
ax.set_xlabel('DT', size = 20, fontname="Times New Roman")
ax.set_ylabel('SGR', size = 20, fontname="Times New Roman")
ax.set_zlabel('SW', size = 20, fontname="Times New Roman")
scatter = ax.scatter3D(XNF, YNF, ZNF,s=70, c='blue',cmap=plt.cm.viridis, marker='o', label='Non_Fracture
Zone')
scatter = ax.scatter3D(XF, YF, ZF,s=70, c='red',cmap=plt.cm.viridis, marker='^', label='Fracture Zone')
plt.show()
ax.legend()
ax.set_xlabel('DT', size = 20, fontname="Times New Roman")
ax.set_ylabel('RHOB', size = 20, fontname="Times New Roman")
```

```
ax.set_zlabel('PEF', size = 20, fontname="Times New Roman")
CGR_RHOB_SGR119.savefig("CGR_RHOB_SGR119.JPEG", bbox_inches='tight', dpi=600)
CGR_SGR_CALI119.savefig("CGR_SGR_CALI119.JPEG", bbox_inches='tight', dpi=600)
DT_RHOB_PEF119.savefig("DT_RHOB_PEF119.JPEG", bbox_inches='tight', dpi=600)
DT_SGR_SW119.savefig("DT_SGR_SW119.JPEG", bbox_inches='tight', dpi=600)
```

Appendix M

Code: Ordered Weighted Averaging

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
import matplotlib.pyplot as pp
import pylab as plot
import matplotlib as mpl
mpl.rc('font',family='Times New Roman')
params = {'legend.fontsize': 27,
          'legend.handlelength': 2
          }
plot.rcParams.update(params)
RFSVM119 = ('D:\Documents\Haleh CSCI-Thesis\Txt Material\WelltoWellresult\WelltoWellbalance\RF-
SVM Balance\\RFSVMB119.xlsx')
x = pd.ExcelFile(RFSVM119)
df119 = x.parse('Sheet1')
X1 = df119.drop('DEPTH', axis=1)
X2 = X1.drop('Fracture', axis=1)
Sortassend119 = np.sort(X2,axis=1)
Sd119 =abs(np.sort(-X2))
Sortdessend119 = pd.DataFrame(Sd119)
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\Sortdessend119.csv',Sortdessend119,
delimiter=',')
Sortdessend119
Sortdessend119.info()
Sortdessend119.index
[r,c] =  Sortdessend119.shape
wo = np.zeros((100, 14))
alpha = np.linspace(0,1,num =100)
for j in range (100):
    for i in range (14):
        if i == 0:
            wo[j, i] = alpha [j]
        elif i < 13:
            wo[j, i] = alpha [j] * (1 - alpha [j]) ** i
        else:
            wo[j, i] = (1 - alpha [j]) ** i
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\OWAo119.csv',wo, delimiter=',')
wp = np.zeros((100, 14))
```

```
alpha = np.linspace(0,1,num =100)
for j in range (100):
    for i in range (14):
        if i == 0:
            wp[j, i] = alpha [j] ** (14-(i+1))
        elif i < 13:
            wp[j, i] = (1 - alpha[j]) * alpha[j]**(14-(i+1))
        else:
            wp[j, i] = 1 - alpha[j]
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\OWAp119.csv',wp, delimiter=',')
wot = np.transpose(wo)
wpt = np.transpose(wp)
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\wot119.csv',wot, delimiter=',')
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\wpt119.csv',wpt, delimiter=',')
reso119 = np.dot(Sortdessend119,wot)
resO119 = np.round(reso119)
[ro,co] = resO119.shape
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\resO119.csv',resO119, delimiter=',')
resp119 = np.dot(Sortdessend119,wpt)
resP119 = np.round(resp119)
[rp,cp] = resP119.shape
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\resP119.csv',resP119, delimiter=',')
real = ('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\real119.xlsx')
x = pd.ExcelFile(real)
real119 = x.parse('Sheet1')
Eo119 = np.subtract(resO119, real119)
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\Eo119.csv',Eo119, delimiter=',')
Ep119 = np.subtract(resP119, real119)
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\Ep119.csv',Ep119, delimiter=',')
ssqo119 = np.sum(Eo119**2)
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\ssqo119.csv',ssqo119, delimiter=',')
ssqp119 = np.sum(Ep119**2)
np.savetxt('D:\Documents\Haleh CSCI-Thesis\Txt Material\OWA\\ssqp119.csv',ssqp119, delimiter=',')
ssqo119.min()
ssqp119.min()
minssqo119 = np.where(ssqo119 == ssqo119.min())
minssqo119
minssqp119 = np.where(ssqp119 == ssqp119.min())
minssqp119
alpha[minssqo119]
alpha[minssqp119]
plt.plot(alpha,ssqo119)
plt.plot(alpha,ssqp119)
sse_alpha119 = plt.figure(figsize=(16,9))
Ploto = plt.scatter(alpha, ssqo119,c='black', alpha=1,linewidths = 2, marker ="o",edgecolor ="black",s = 100)
Plotp = plt.scatter(alpha, ssqp119,c='black', alpha=1, linewidths = 3, marker ="x",edgecolor ="black", s = 100)
import matplotlib.pyplot as plt
plt.xlim(0,1)
plt.ylim(0,250)
plt.legend([Ploto, Plotp], ['Optimistic OWA ','Pessimistic OWA '])
plt.xlabel('Alpha', size = 27, fontname="Times New Roman")
plt.ylabel('SSE ', size = 27, fontname="Times New Roman")
plt.xticks(size = 25, fontname="Times New Roman")
```

```
plt.yticks(size = 25, fontname="Times New Roman")
sse_alpha119.savefig("SSE_ALPHAbalance119.JPEG", bbox_inches='tight', dpi=600)
```

Appendix N

Code: ROC (Receiver Operating Characteristic) Curve for SVM Classifier

```
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from IPython.display import display
from IPython.display import Image
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x119 = pd.ExcelFile(path119)
df119 = x119.parse('Sheet1')
X1 = df119.drop('CALI', axis=1)
X119 = X1.drop(['DOLOMITE','SHALE','LIME','PORE','SW'], axis=1 )
path245 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2Balanced\\GS#245.xlsx')
path = pd.ExcelFile(path245)
df245 = path.parse('Sheet1')
zeros = df245.loc[(df245['Fracture'] == 0)].sample(n = 1450)
ones = df245.loc[(df245['Fracture'] == 1)]
dfbalance245 = pd.concat([zeros, ones])
X2 = dfbalance245.drop('CALI', axis=1)
X245 = X2
X_train = X245.drop('Fracture', axis=1)
y_train = X245['Fracture']
X_test = X119.drop('Fracture', axis=1)
y_test = X119['Fracture']
X_train, X_test, y_train, y_test = train_test_split(pd.concat([X119, X245]).drop('Fracture', axis=1),
pd.concat([X119, X245])['Fracture'], test_size=0.3)
model119245 = SVC()
model119245.fit(X_train,y_train)
predictions119245 = model119245.predict(X_test)
param_grid = {'C':[0.1,1,10,100,1000],'gamma':[1,0.1,0.01,0.001,0.0001]}
grid119245 = GridSearchCV(SVC(),param_grid,verbose=3)
grid119245.fit(X_train,y_train)
grid_predictions119245 = grid119245.predict(X_test)
SVM119245 = grid119245.predict(X119.drop('Fracture', axis=1))
```

```
print(confusion_matrix(X119['Fracture'],SVM119245))
print('\n')
print(classification_report(X119['Fracture'],SVM119245))
print ('\n')
print("Accuracy:", accuracy_score(X119['Fracture'], SVM119245))
res = pd.DataFrame(SVM119245)
res.columns = ["SVM119245"]
res.to_csv("D:\Documents\Haleh CSCI-Thesis\Txt
Material\WelltoWellresult\WelltoWellbalance\\SVM119245.csv")
from sklearn import metrics
preds = model119245.predict(X_train)
targs = (y_train)
print("accuracy: ", metrics.accuracy_score(targs, preds))
print("precision: ", metrics.precision_score(targs, preds))
print("recall: ", metrics.recall_score(targs, preds))
print("f1: ", metrics.f1_score(targs, preds))
print("area under curve (auc): ", metrics.roc_auc_score(targs, preds))
train_preds = preds
preds = model119245.predict(X_test)
targs = (y_test)
print("accuracy: ", metrics.accuracy_score(targs, preds))
print("precision: ", metrics.precision_score(targs, preds))
print("recall: ", metrics.recall_score(targs, preds))
print("f1: ", metrics.f1_score(targs, preds))
print("area under curve (auc): ", metrics.roc_auc_score(targs, preds))
test_preds = preds
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import roc_curve, auc
y_train_pred =model119245.decision_function(X_train)
y_test_pred = model119245.decision_function(X_test)
train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
ROC119245 = plt.figure(figsize=(16,9))
plt.grid()
plt.plot(train_fpr, train_tpr,linewidth=3, label=" AUC TRAIN ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, linewidth=3, label=" AUC TEST ="+str(auc(test_fpr, test_tpr)))
plt.rcParams["legend.fontsize"] = 20
plt.plot([0,1],[0,1],'g--' , linewidth=3)
plt.legend()
plt.xlabel("Specifity(Fals Positive Rate)",size = 22, fontname="Times New Roman")
plt.ylabel("Sensitivity(True Positive Rate)",size = 22, fontname="Times New Roman")
plt.title("AUC(ROC curve)",size = 20, fontname="Times New Roman")
plt.grid(color='black', linestyle='-', linewidth=1)
plt.show()
```

Appendix O

Code: ROC (Receiver Operating Characteristic) Curve for RF Classifier

```
import numpy as np
import pandas as pd
import xlsxwriter
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('classic')
get_ipython().run_line_magic('matplotlib', 'inline')
import pylab as plot
import matplotlib as mpl
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from IPython.display import display
from IPython.display import Image
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
path119 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2\\GS#119.xlsx')
x119 = pd.ExcelFile(path119)
df119 = x119.parse('Sheet1')
X1 = df119.drop('CALI', axis=1)
X119 = X1.drop(['DOLOMITE','SHALE','LIME','PORE','SW'], axis=1 )
path245 = ('D:\Documents\Haleh CSCI-Thesis\Data (Haleh)\Final Data For
Start\Petrophysics\Petrophysics2Balanced\\GS#245.xlsx')
path = pd.ExcelFile(path245)
df245 = path.parse('Sheet1')
zeros = df245.loc[(df245['Fracture'] == 0)].sample(n = 1450)
ones = df245.loc[(df245['Fracture'] == 1)]
dfbalance245 = pd.concat([zeros, ones])
X2 = dfbalance245.drop('CALI', axis=1)
X245 = X2
X_train = X245.drop('Fracture', axis=1)
y_train = X245['Fracture']
X_test = X119.drop('Fracture', axis=1)
y_test = X119['Fracture']
X_train, X_test, y_train, y_test = train_test_split(pd.concat([X119, X245]).drop('Fracture', axis=1),
pd.concat([X119, X245])['Fracture'], test_size=0.3)
rfc119245 = RandomForestClassifier(n_estimators=500)
rfc119245.fit(X_train, y_train)
rfc_pred119245 = rfc119245.predict(X_test)
print(confusion_matrix(y_test,rfc_pred119245 ))
print('\n')
print(classification_report(y_test,rfc_pred119245 ))
print ('\n')
```

```
print("Accuracy:", accuracy_score(y_test, rfc_pred119245))
import pylab as pl
from sklearn.utils import shuffle
from sklearn.metrics import roc_curve, auc
random_state = np.random.RandomState(0)
#classifier = svm.SVC(kernel='linear', probability=True)
probas_ = rfc119245.fit(X_train, y_train).predict_proba(X_test)
# Compute ROC curve and area the curve
fpr, tpr, thresholds = roc_curve(y_test, probas_[:, 1])
roc_auc = auc(fpr, tpr)
print( "Area under the ROC curve : %f" % roc_auc)
# Plot ROC curve
ROC119245 = plt.figure(figsize=(16,9))
pl.clf()
plt.grid()
pl.plot(fpr, tpr, linewidth=3, label='ROC curve (area = %0.2f)' % roc_auc)
plt.rcParams["legend.fontsize"] = 20
pl.plot([0, 1], [0, 1], 'g--', linewidth=3)
pl.xlim([0.0, 1.0])
pl.ylim([0.0, 1.0])
pl.xlabel('Specifity (Fals Positive Rate)',size = 22, fontname="Times New Roman")
pl.ylabel('Sensitivity (True Positive Rate)',size = 22, fontname="Times New Roman")
pl.title('AUC(ROC curve)',size = 22, fontname="Times New Roman")
pl.legend(loc="lower right")
pl.show()
```